



# Основы маршрутизации в Интернет

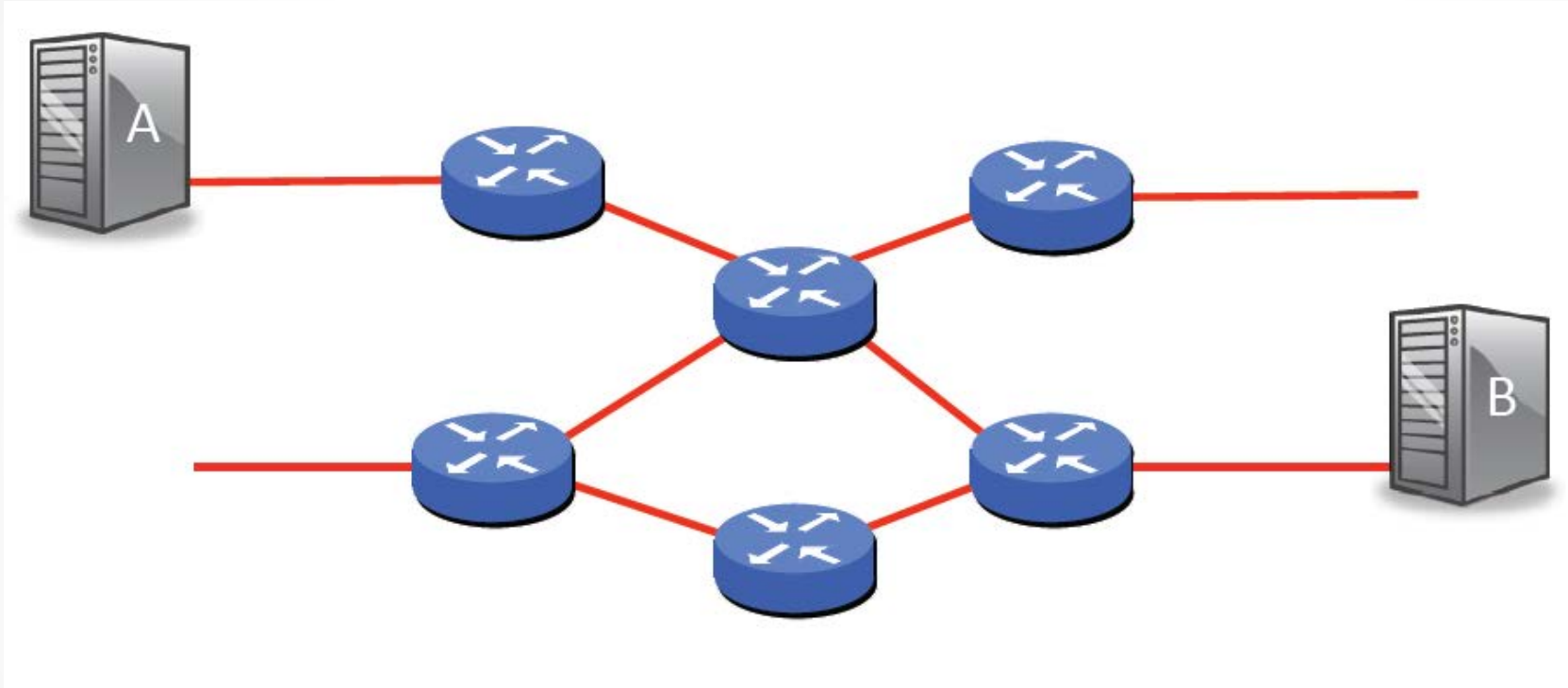
(том 2 стр.30-52)

Введение в компьютерные сети  
чл.-корр. РАН Смелянский Р.Л.  
Кафедра АСВК  
ф-т ВМК МГУ





# Проблема

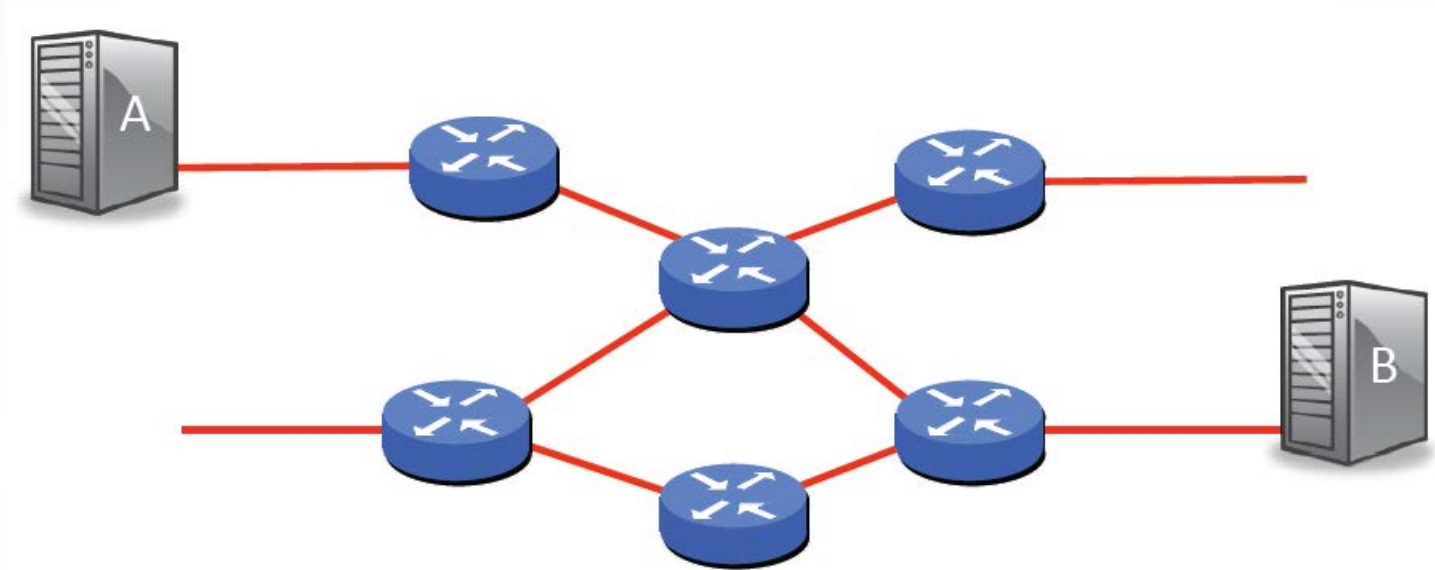


- Кто должен определить как пакеты из A достигнут B?

Критерий выбора маршрута?

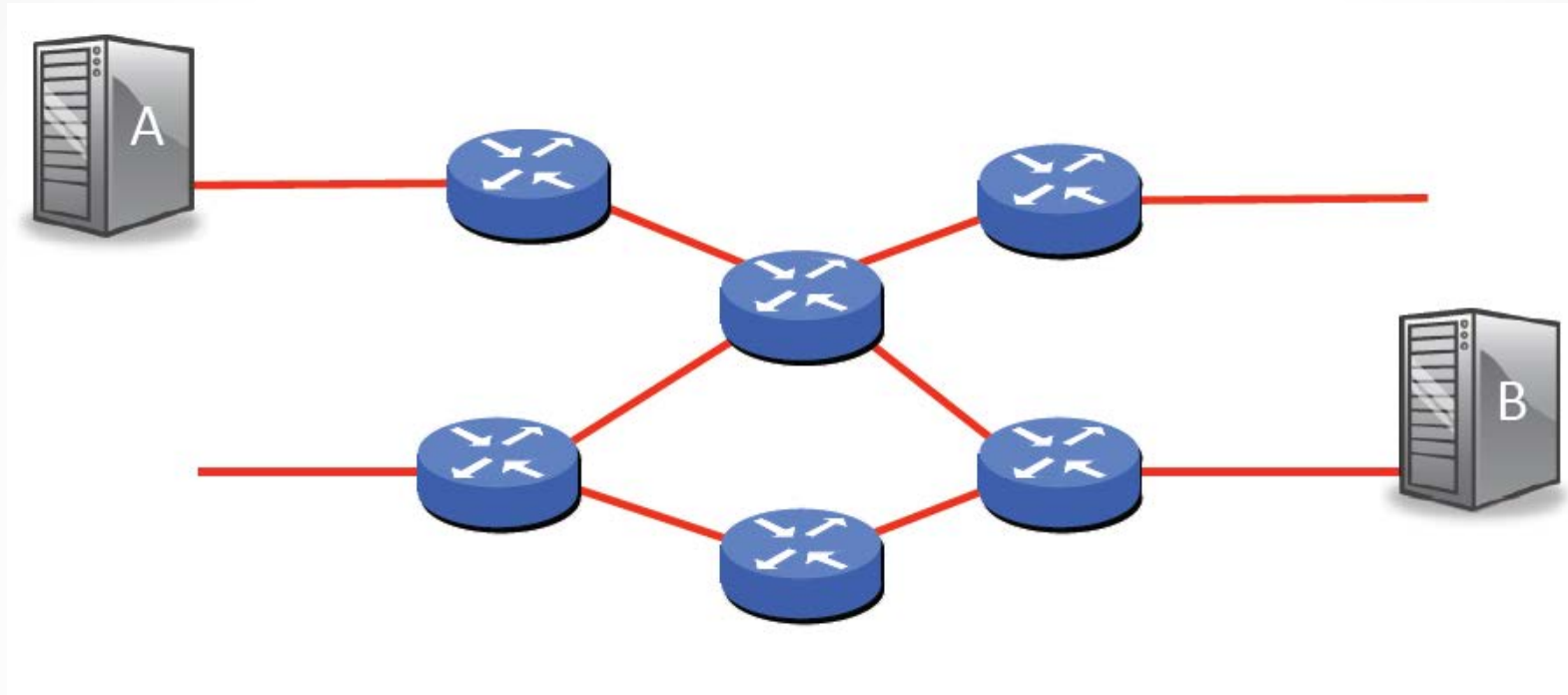


# Лавина (Flooding)



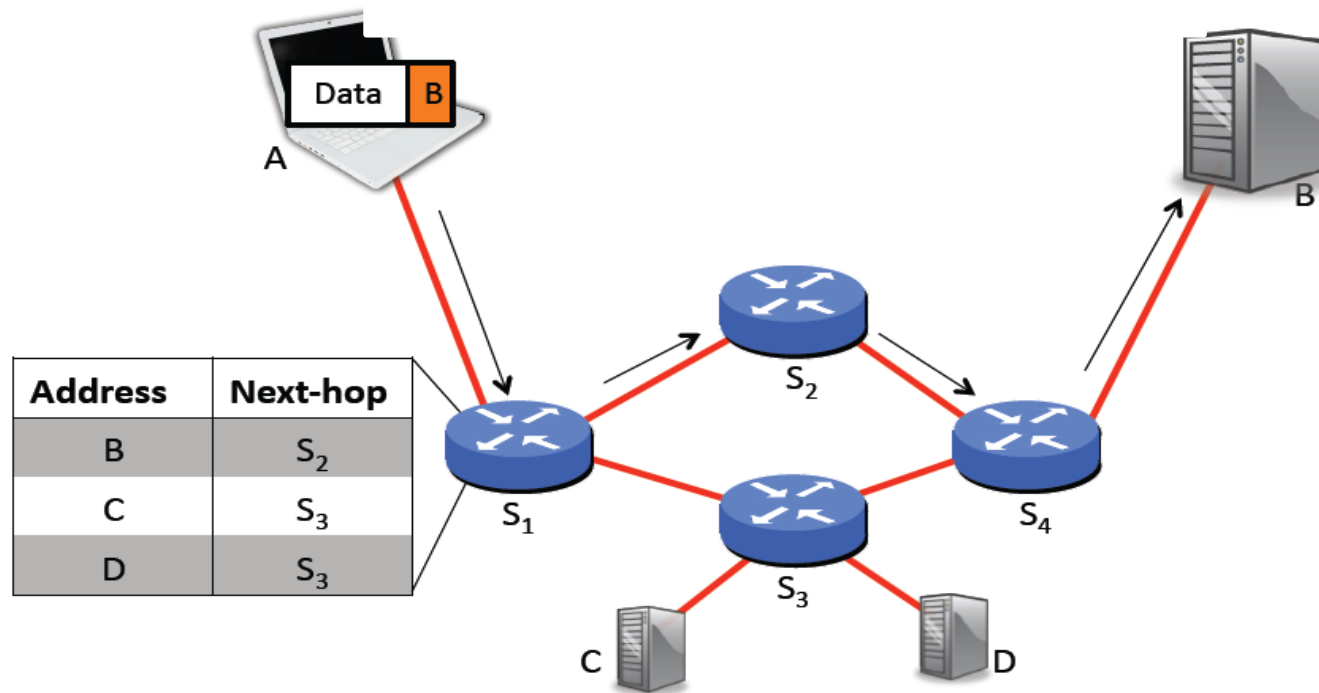
- Не эффективное использование линий
- Нагрузка на сеть
- Пакеты могут зацикливаться
- Как отличить оригинал от дубля?
- Используется когда топология не известна (или ей нельзя доверять)

# Маршрутизация от источника



- Не требует поддержки от сети (маршрутизаторы только коммутуют)
- Пакеты содержат списки адресов, переменной длины (могут быть очень длинными)
- Выбор маршрута на конечном хосте, который должен знать топологию сети
- Используется когда пользователь хочет сам управлять маршрутизацией

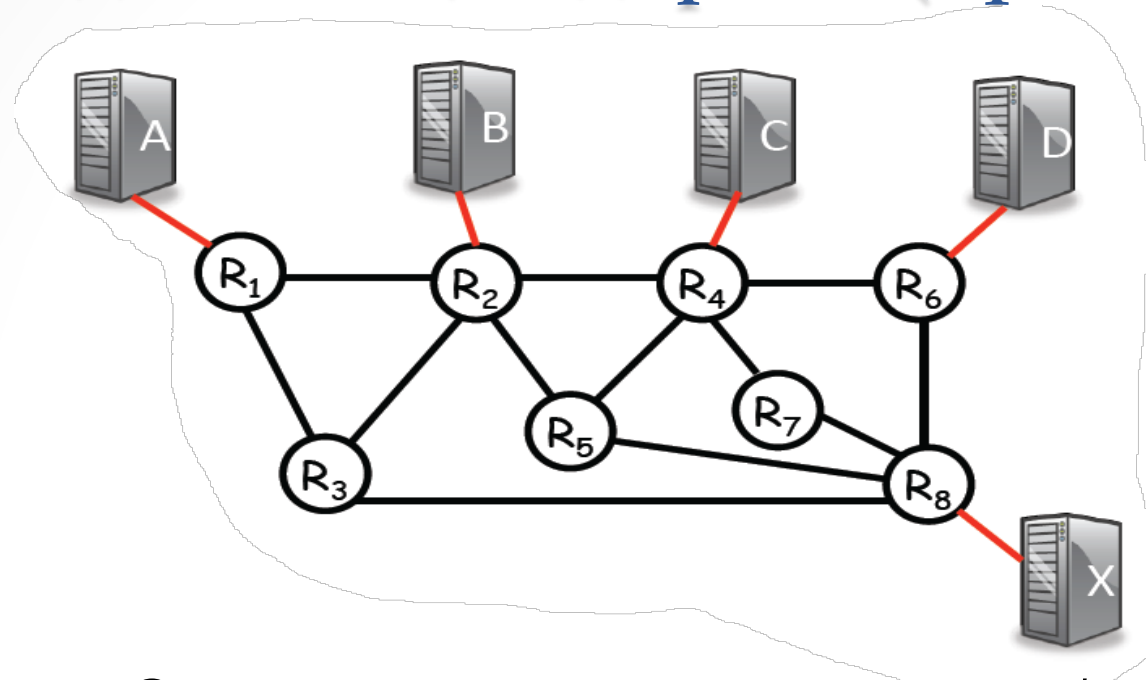
# Таблицы маршрутизации



- Оптимизация: сеть маршрутизирует по скачкам
- У каждого маршрутизатора должна быть своя таблица (необходимо много таблиц)
- Значения полей таблицы зависят от места назначения, а не от потока
- Как сформировать и поддерживать таблицу в актуальном состоянии?



# Соединяющее дерево (Spanning tree)



Соединяющее дерево – остов графа

**ST - граф**, без циклов и состоящий из ребер графа  $G$  и всех его вершин.

**ST - остов графа  $G = (V, E)$ :  $ST = (V, E')$ , где  $E' \subset E$**

$|E'| = \text{rang } G$  (число ребер остова равно рангу графа);

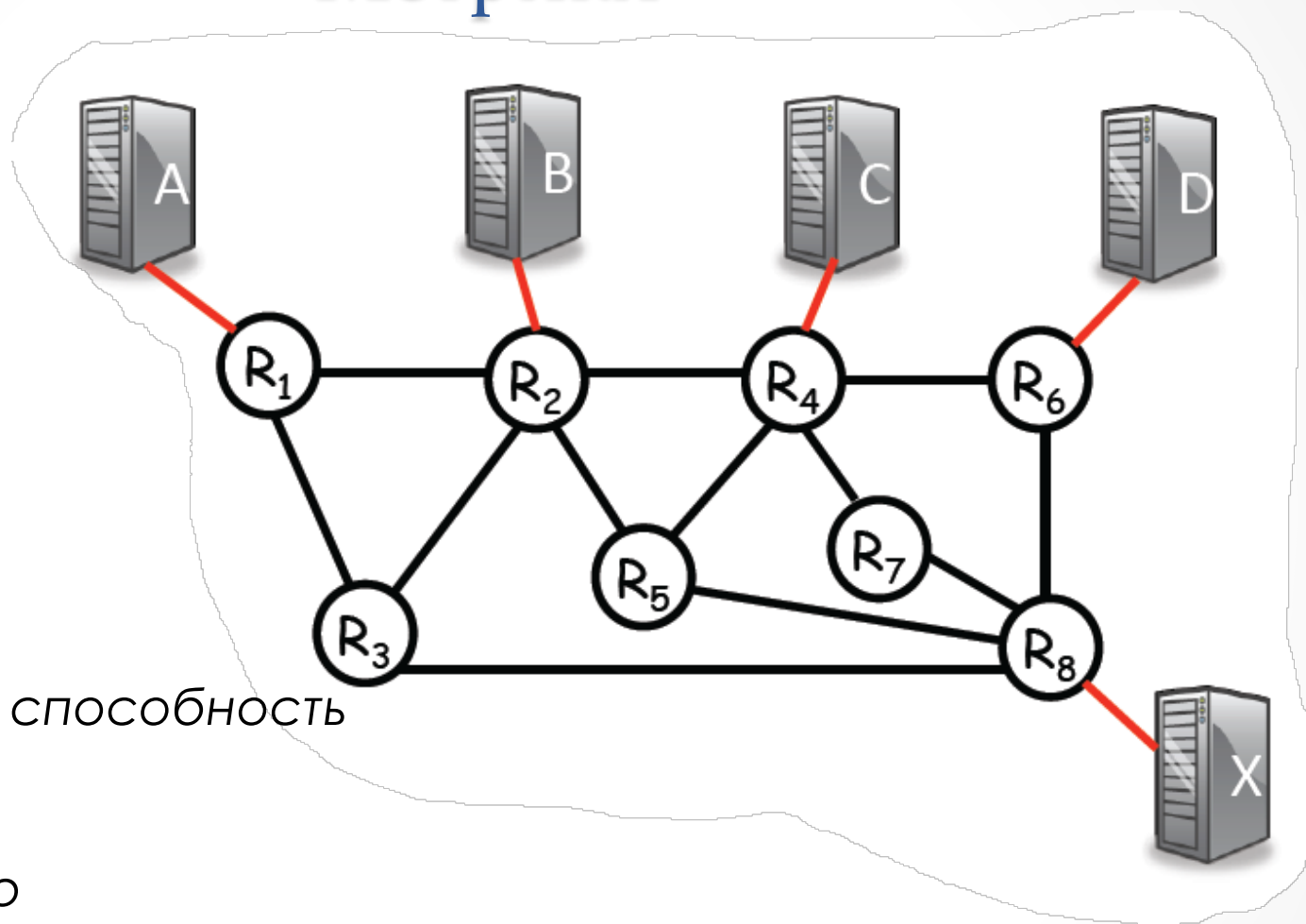
$\text{rang } G$  – максимальный не нулевой минор матрицы смежности  $G$ .



# Метрики

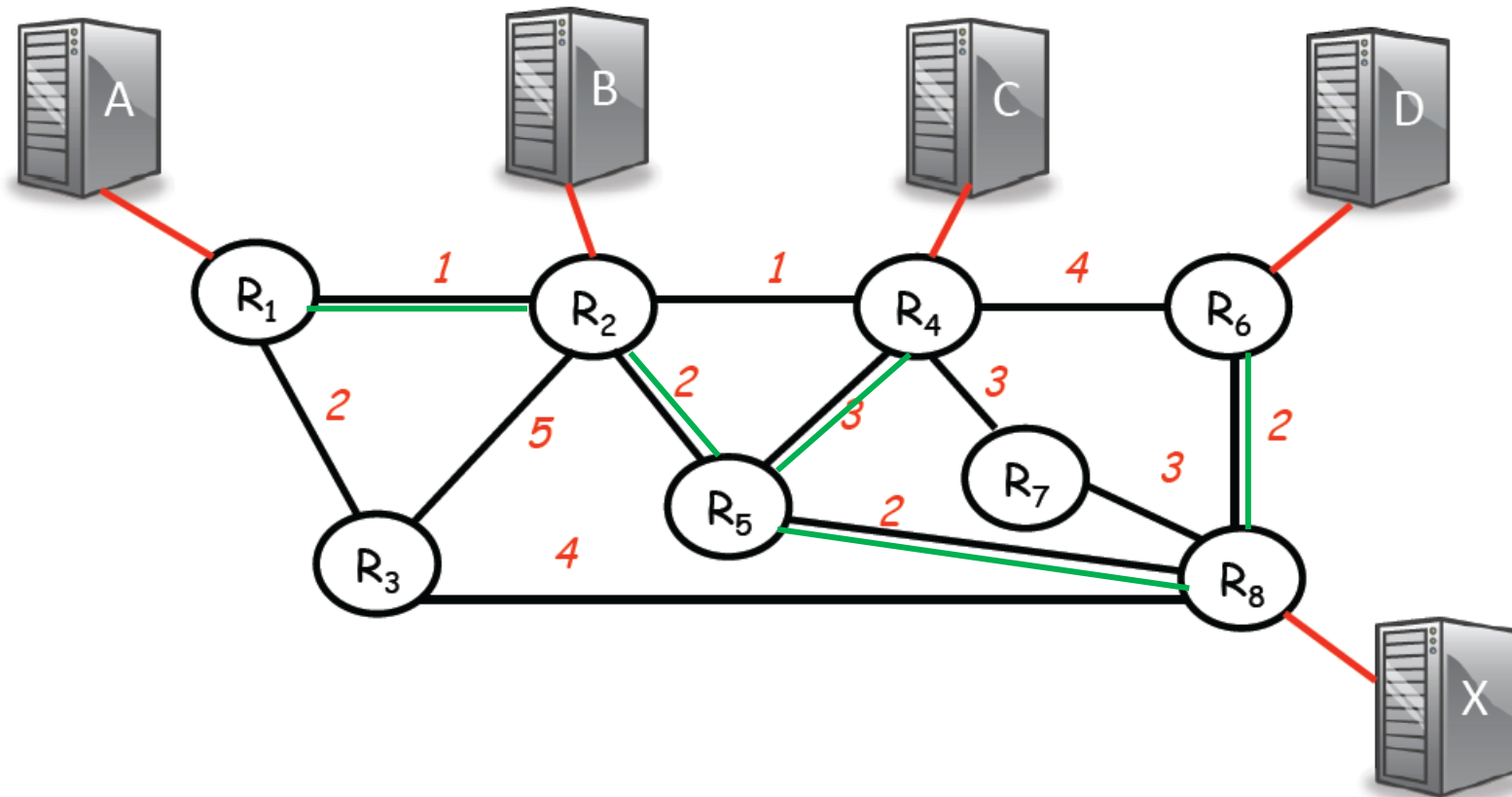
Метрики:

- мин. расстояние
- мин. скачки
- мин. задержка
- макс. пропускная способность
- мин. загруженный
- макс. надежный
- с мин. стоимостью
- макс. безопасный
- ...





# Пример взвешенного графа

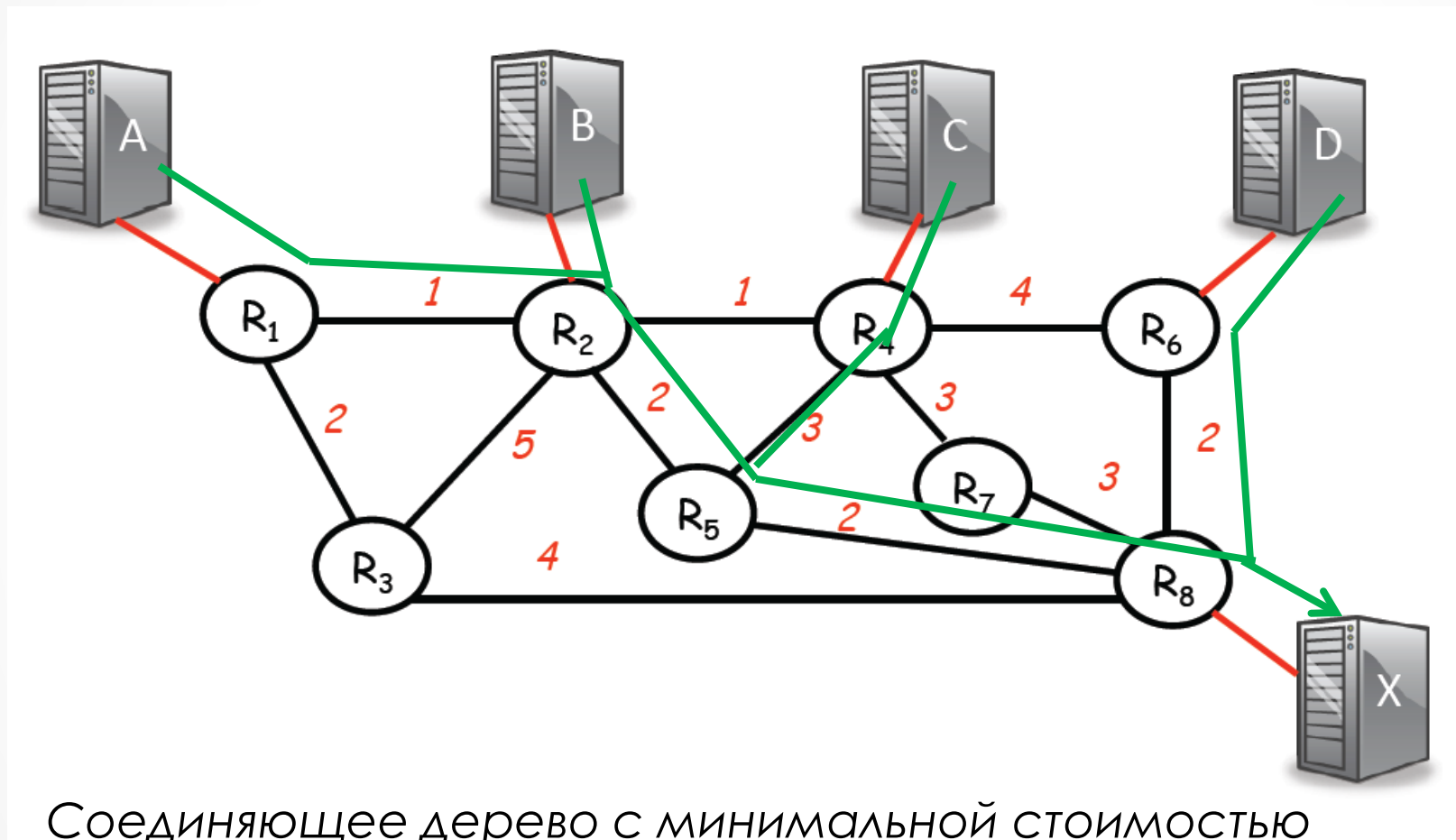




# Понятие взвешенного графа

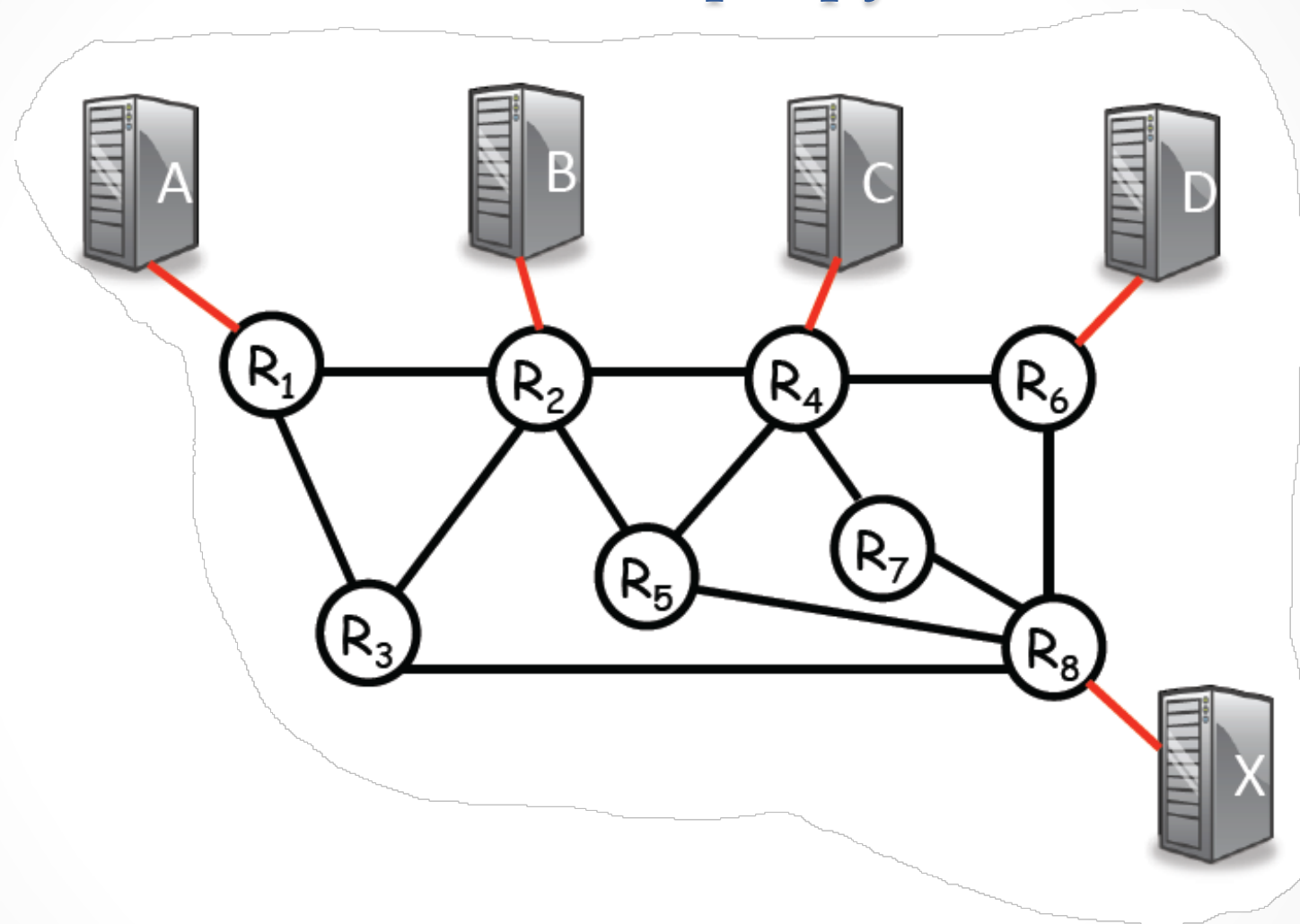


Соединяющее дерево: корень отправитель, листья – все достижимые хосты.



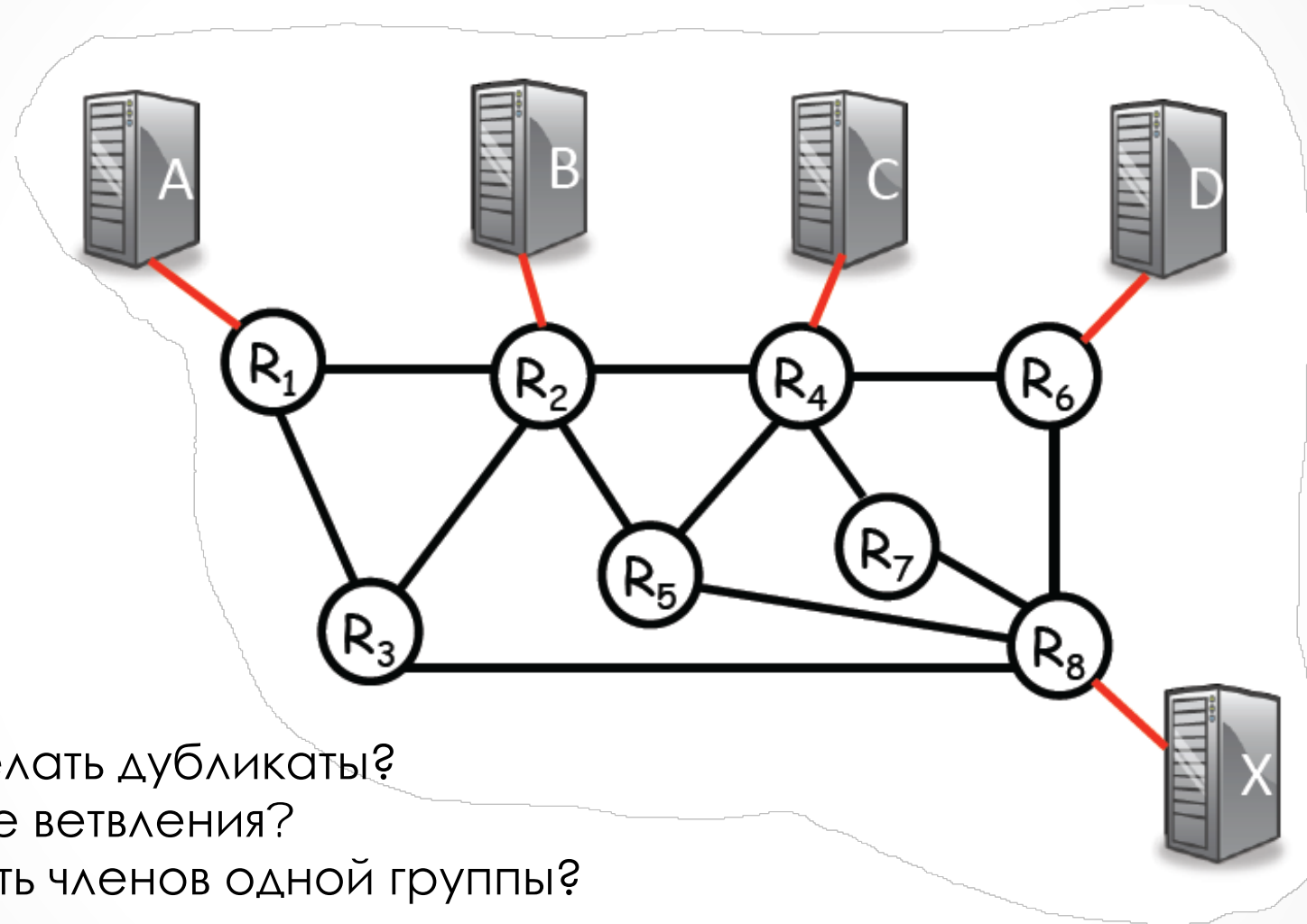


# Многопоточная маршрутизация





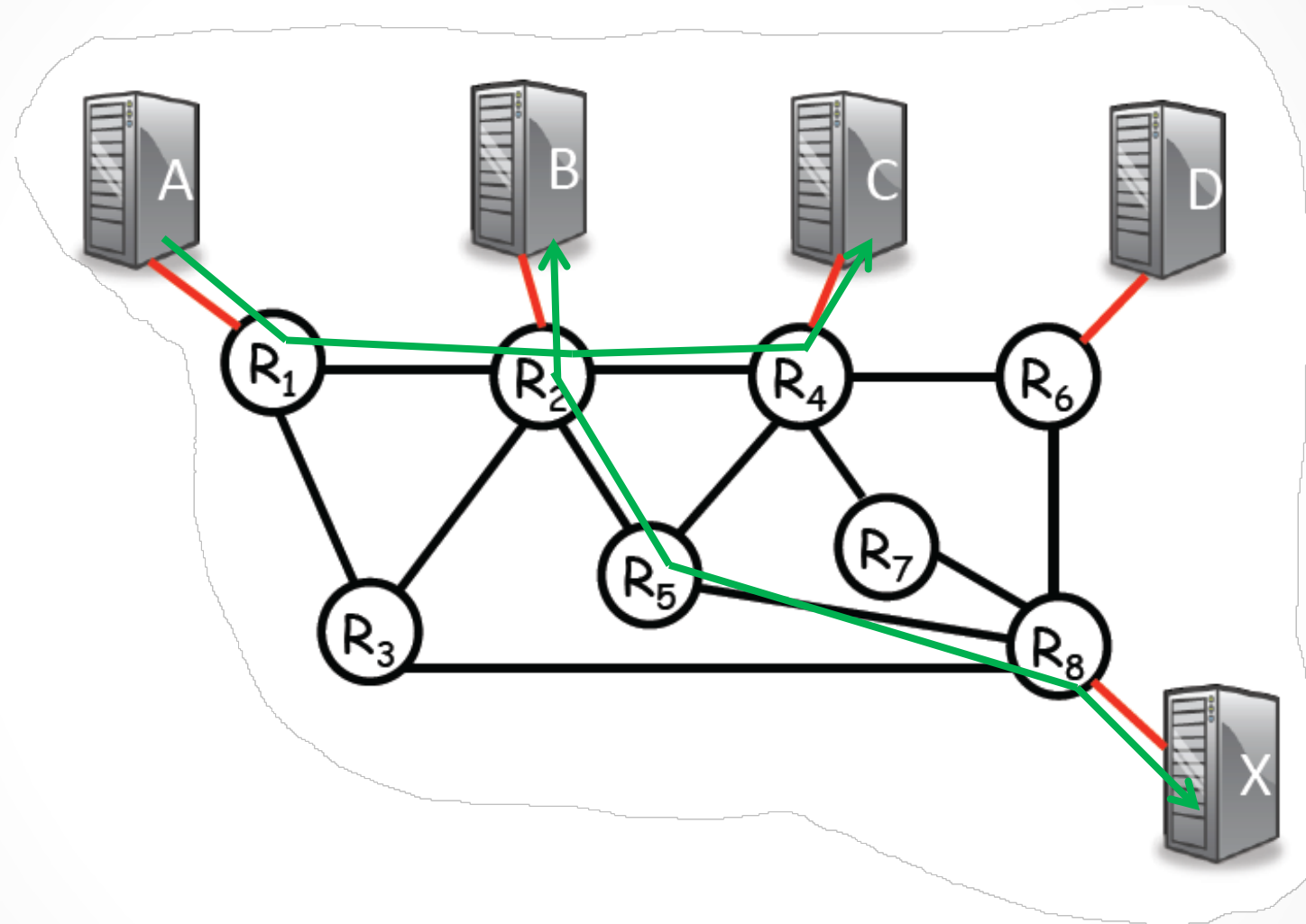
# Групповая



- Кто должен делать дубликаты?
- В каждой точке ветвления?
- Как определить членов одной группы?



# Групповая





# Маршрутизация по вектору расстояния: алгоритм Белмана-Форда (том 2 стр.38-41)

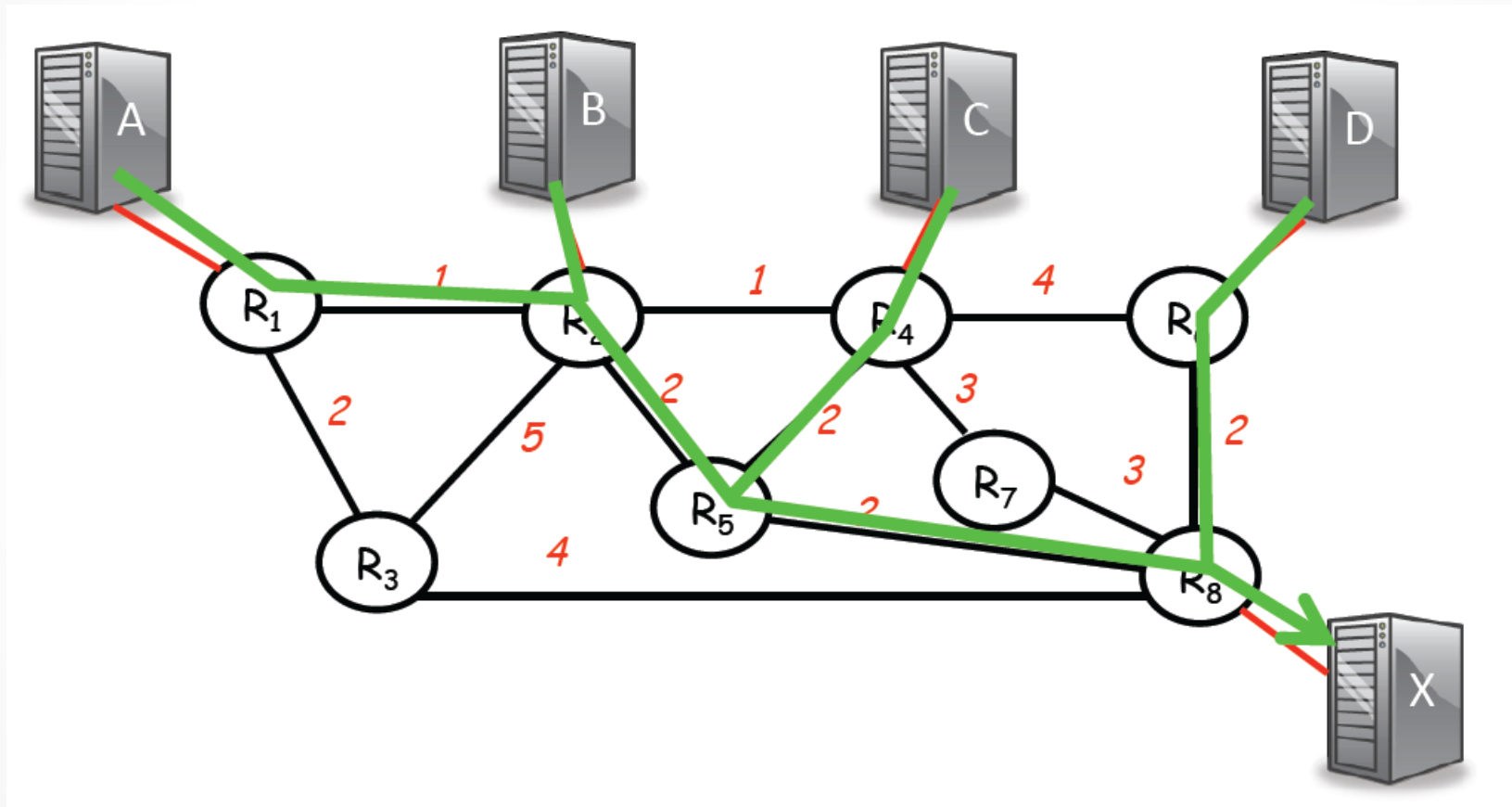
Введение в компьютерные сети  
чл.-корр. РАН Смелянский Р.Л.  
Кафедра АСВК  
ф-т ВМК МГУ





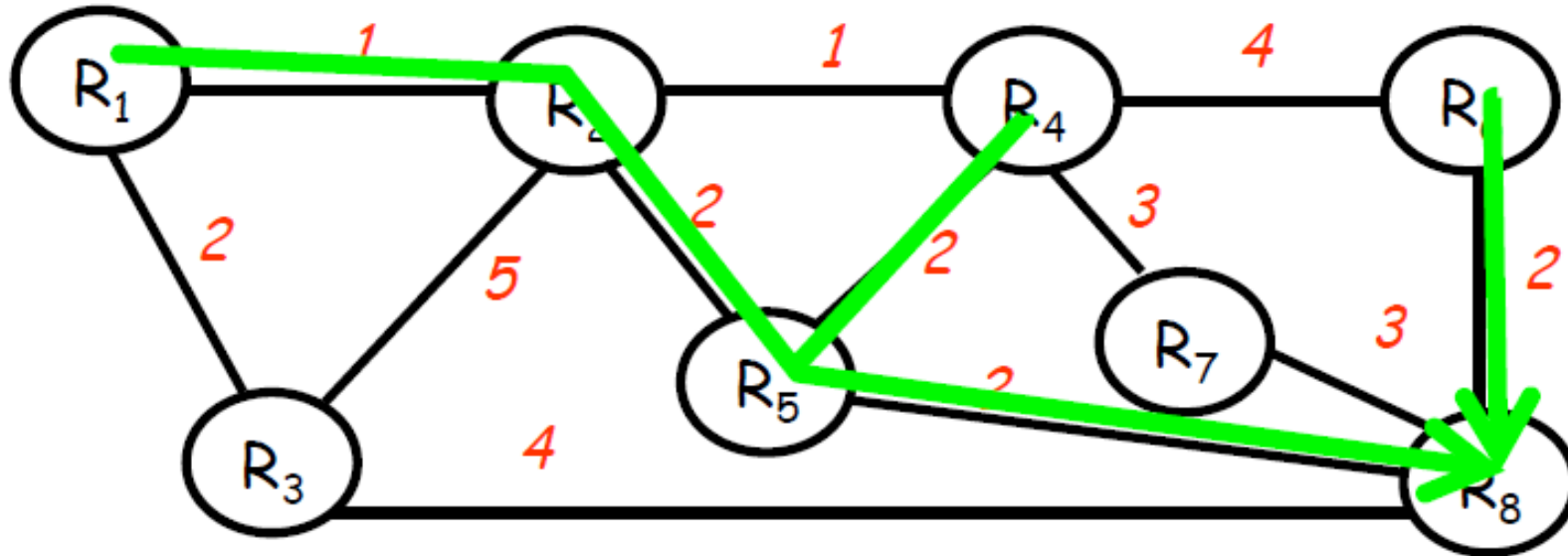
# Проблема

Как маршрутизаторы могут совместно найти соединяющее дерево минимальной стоимости?





Эквивалентно нахождению соединяющего дерева минимальной стоимости только среди маршрутизаторов





# Распределенный алгоритм Беллмана-Форда

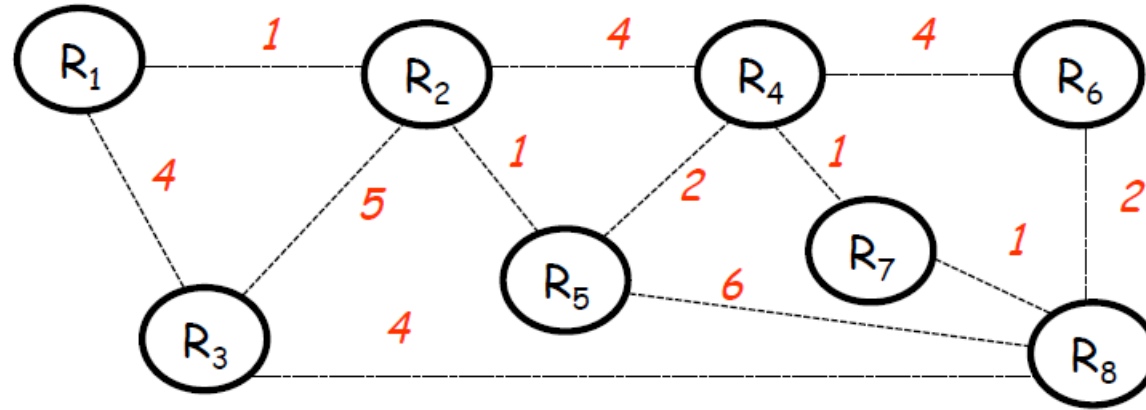
(т.2 стр.38-48)

- Пусть каждый маршрутизатор знает стоимость линии к каждому своему непосредственному соседу
- Маршрутизатор  $R_8$  рассчитывает стоимость  $C_i$  для достижения каждого известного ему  $R_i$
- Вектор  $\underline{C}_8 = (C_1, C_2, \dots, C_7)$  - вектор расстояния до  $R_8$
- Изначально  $\underline{C} = (\infty, \infty, \dots, \infty)$ 
  1. Каждые  $T$  секунд,  $R_i$  шлет  $C_i$  всем своим соседям
  2. Если  $R_i$  нашел более дешевый путь, то он обновляет  $C_i$  у всех своих соседей
  3. Вернуться к 1
- Длину вектора  $C$  устанавливает администратор сети





# Пример



$R_1$	$\infty$	$R_1$	$\infty$	$R_1$	$8, R_3$	$R_1$	$8, R_3$	$R_1$	$7, R_2$	$R_1$	$6, R_3$
$R_2$	$\infty$	$R_2$	$\infty$	$R_2$	$7, R_5$	$R_2$	$6, R_4$	$R_2$	$5, R_7$	$R_2$	$5, R_7$
$R_3$	$\infty$	$R_3$	$4$	$R_3$	$4$	$R_3$	$4$	$R_3$	$4$	$R_3$	$4$
$R_4$	$\infty$	$R_4$	$\infty$	$R_4$	$2, R_7$	$R_4$	$2, R_7$	$R_4$	$2, R_7$	$R_4$	$2, R_7$
$R_5$	$\infty$	$R_5$	$6$	$R_5$	$6$	$R_5$	$4, R_4$	$R_5$	$4, R_4$	$R_5$	$4, R_4$
$R_6$	$\infty$	$R_6$	$2$	$R_6$	$2$	$R_6$	$2$	$R_6$	$2$	$R_6$	$2$
$R_7$	$\infty$	$R_7$	$1$	$R_7$	$1$	$R_7$	$1$	$R_7$	$1$	$R_7$	$1$
<b>шаг 0</b>		<b>шаг 1</b>		<b>шаг 2</b>		<b>шаг 3</b>		<b>шаг 4</b>		<b>шаг 5</b>	



# Алгоритм Беллмана-Форда

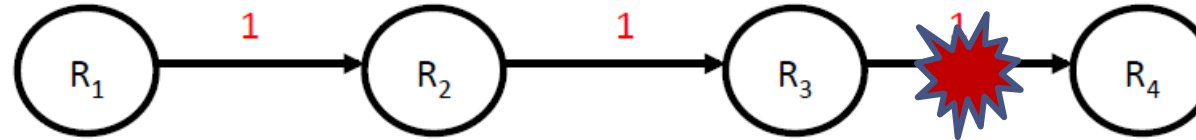
*Вопросы:*

- 1. Каково максимальное время работы алгоритма?*
- 2. Всегда ли алгоритм будет сходиться?*
- 3. Что будет если измениться стоимость линии, или отключится маршрутизатор/линия?*



# Проблемы с алгоритмом Б-Ф

Данные об отказе распространяются медленно



Рассмотрим расчет расстояния для  $R_3$  до  $R_4$

Time	$R_1$	$R_2$	$R_3$
0	3, $R_2$	2, $R_3$	1, $R_4$
1	3, $R_2$	2, $R_3$	3, $R_2$
2	3, $R_2$	4, $R_3$	3, $R_2$
3	5, $R_2$	4, $R_3$	5, $R_2$
...	Итак до бесконечности		...

$R_4$

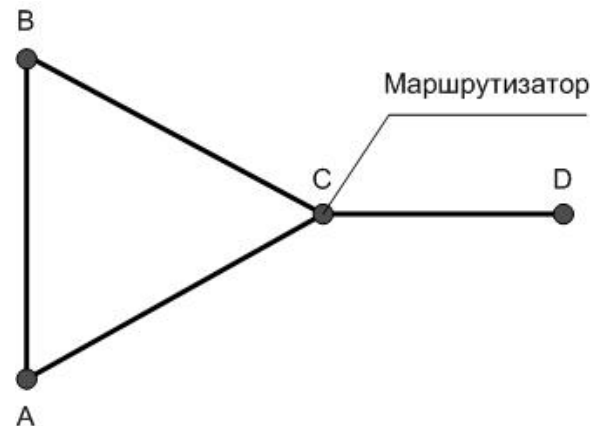
Линия  $R_3 - R_4$  не действует.  
 $R_2: (1, 0, 1, 2) \Rightarrow R_3: (2, 1, 0, 3)$



# Проблема счетчика до бесконечности

Установить ограничение на «бесконечность» (e.g. 16)

- *Разделение направлений: т.к.  $R_2$  получает данные о маршруте с наименьшей стоимостью от  $R_3$ , то запретить  $R_2$  сообщать  $R_3$  о маршрутах, проходящих через  $R_3$*
- *Разделение направлений с бесконечностью:  $R_2$  посылает  $R_3$   $\infty$*
- *Есть и другие проблемы, связанные с алгоритмом Б-Ф*



**Проблема в том, что топологию сети маршрутизаторы не знают!**



## Беллман-Форд на практике

- У алгоритма Беллмана-Форда низкая сложность, хорошо распараллеливается;
- не требует знания топологии сети, можно добиться сходимости, наложив ограничение на рост счетчика пути, что требует либо ручного вмешательства администратора, либо расчета топологии;
- Этот алгоритм использовался в первых Интернет протоколах маршрутизации RIP (Routing Internet Protocol)
- Со временем он был вытеснен алгоритмами расчета остовного дерева каждым маршрутизатором.



# Маршрутизация по состоянию канала: алгоритм наикратчайшего пути Дейкстры (том 2 стр.33, 41-46)

Введение в компьютерные сети  
чл.-корр. РАН Смелянский Р.Л.

Кафедра АСВК  
ф-т ВМК МГУ



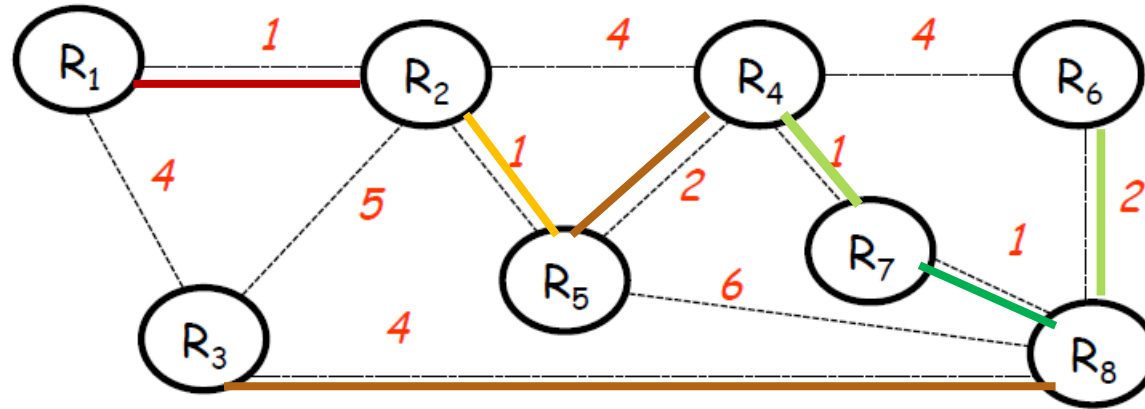


# Алгоритм Дейкстры наикратчайшего пути

1. *Определение топологии сети: каждый маршрутизатор передает лавиной всем своим соседям состояния своих линий;*
2. *На основе полученных данных строит топологию сети*
  - *Периодически*
  - *Когда изменяется состояние линии*
3. *Каждый маршрутизатор строит ST минимального веса независимо от других с помощью алгоритма Дейкстры наикратчайшего пути.*



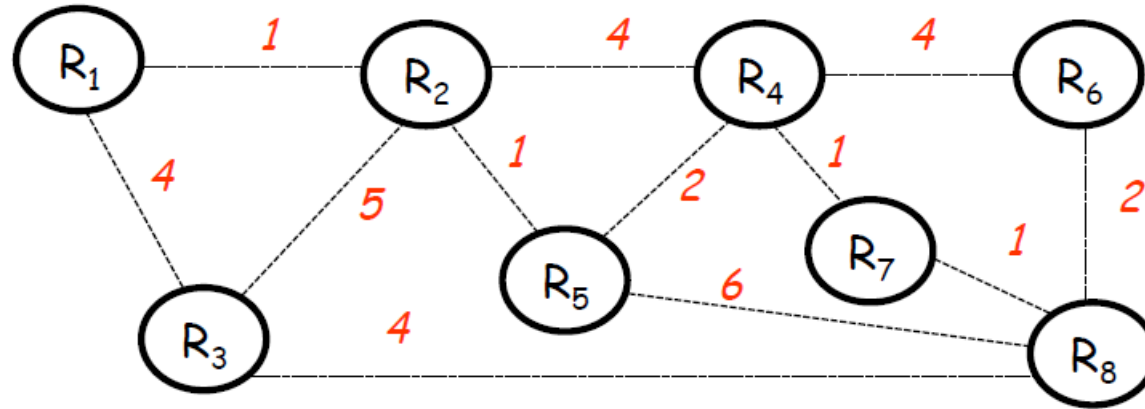
# Пример для $R_8$







# Пример для $R_8$

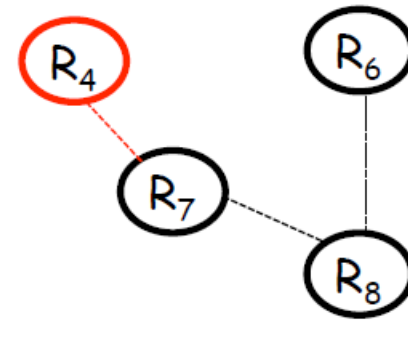
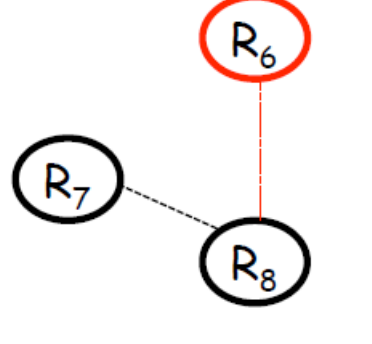
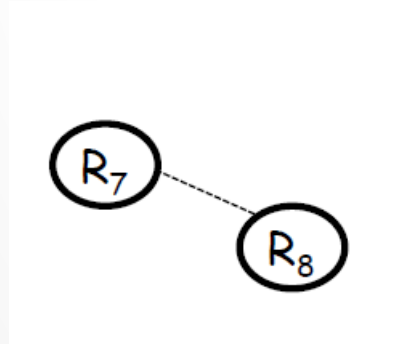


Добавляем путь стоимости

1

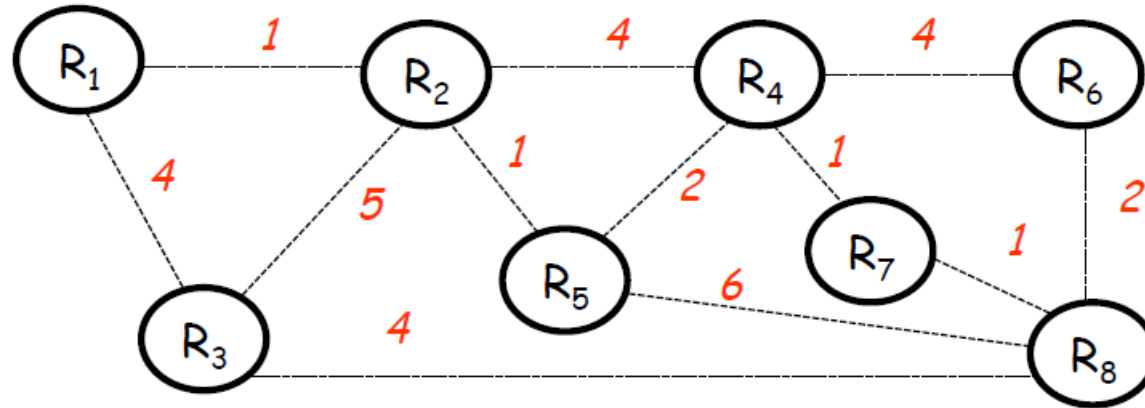
2

2





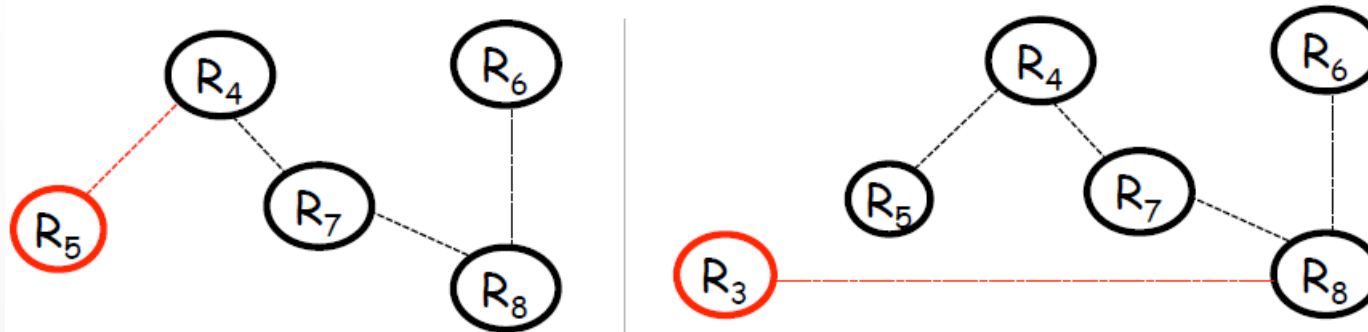
# Пример для $R_8$



Добавляем путь стоимости

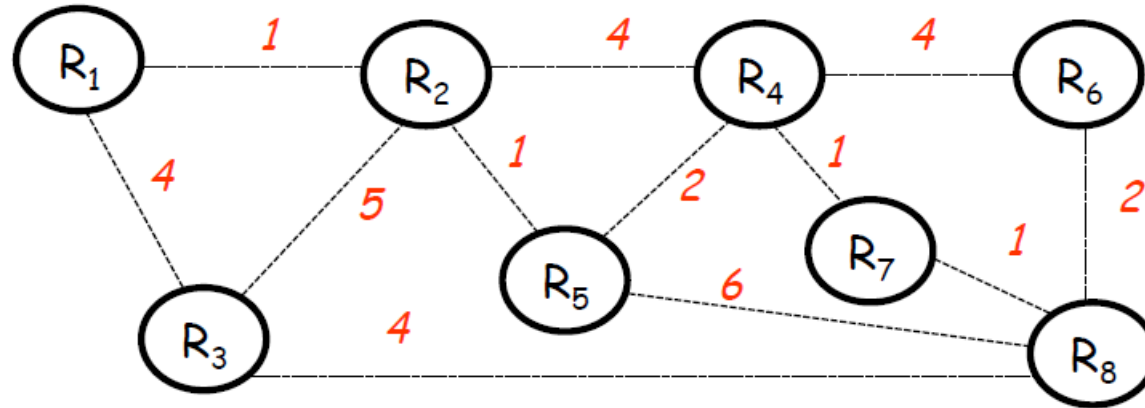
4

4

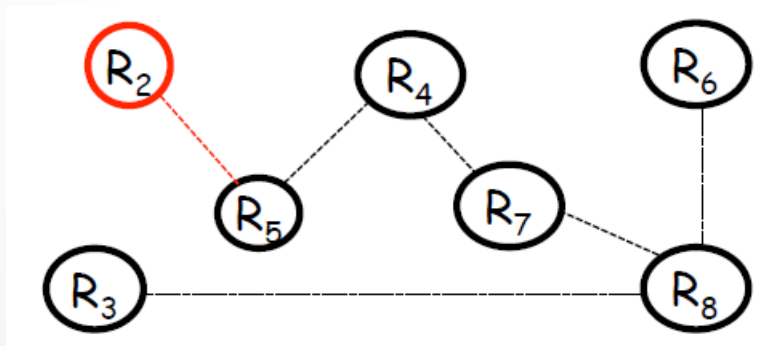




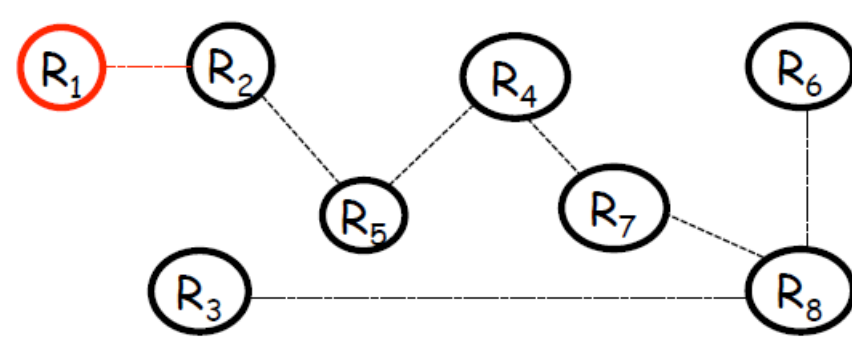
# Пример для $R_8$



Добавляем путь стоимости  
5

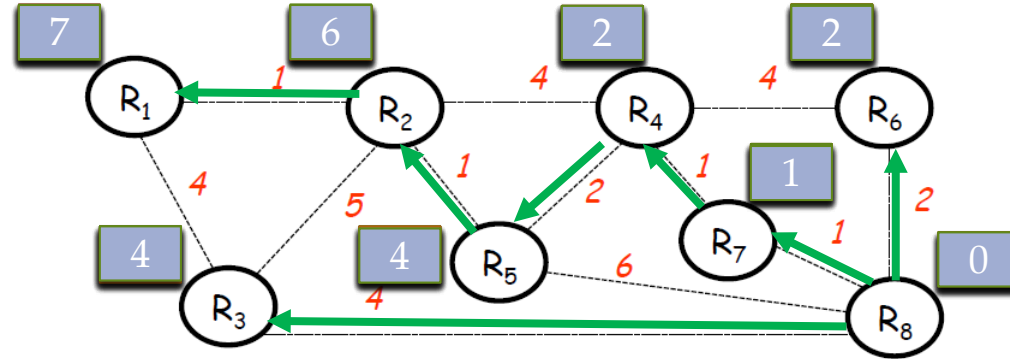


6





# Алгоритм



	0	1	2	3	4	5	6	7
Пройден		R <sub>8</sub>	R <sub>8</sub> , R <sub>7</sub>	R <sub>8</sub> , R <sub>7</sub> , R <sub>6</sub>	R <sub>8</sub> , R <sub>7</sub> , R <sub>6</sub> , R <sub>4</sub>	R <sub>8</sub> , R <sub>7</sub> , R <sub>6</sub> , R <sub>4</sub> , R <sub>3</sub>	R <sub>8</sub> , R <sub>7</sub> , R <sub>6</sub> , R <sub>4</sub> , R <sub>3</sub> , R <sub>5</sub>	R <sub>8</sub> , R <sub>7</sub> , R <sub>6</sub> , R <sub>4</sub> , R <sub>3</sub> , R <sub>5</sub> , R <sub>2</sub>
Весы	R <sub>1</sub> =∞ R <sub>5</sub> =∞ R <sub>2</sub> =∞ R <sub>6</sub> =∞ R <sub>3</sub> =∞ R <sub>7</sub> =∞ R <sub>4</sub> =∞ R <sub>8</sub> =0	R <sub>1</sub> =∞ R <sub>5</sub> =6 R <sub>2</sub> =∞ R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>4</sub> =∞ R <sub>8</sub> =0	R <sub>1</sub> =∞ R <sub>5</sub> =6 R <sub>2</sub> =∞ R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>4</sub> =2 R <sub>8</sub> =0	R <sub>1</sub> =∞ R <sub>5</sub> =6 R <sub>2</sub> =∞ R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>4</sub> =2 R <sub>8</sub> =0	R <sub>1</sub> =∞ R <sub>5</sub> =4 R <sub>2</sub> =6 R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>4</sub> =2 R <sub>8</sub> =0	R <sub>1</sub> =8 R <sub>5</sub> =4 R <sub>2</sub> =6 R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>4</sub> =2 R <sub>8</sub> =0	R <sub>1</sub> =8 R <sub>4</sub> =2 R <sub>2</sub> =5 R <sub>6</sub> =2 R <sub>5</sub> =4 R <sub>7</sub> =1 R <sub>3</sub> =4 R <sub>8</sub> =0	R <sub>1</sub> =6 R <sub>4</sub> =2 R <sub>2</sub> =5 R <sub>6</sub> =2 R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>5</sub> =4 R <sub>8</sub> =0
Смежны	R <sub>3</sub> =4 R <sub>7</sub> =1 R <sub>5</sub> =6 R <sub>6</sub> =2	R <sub>4</sub> =2 R <sub>5</sub> =4 R <sub>6</sub> =2 R <sub>5</sub> =6	R <sub>4</sub> =2 < 6 => R <sub>4</sub> =2 R <sub>3</sub> =4 R <sub>5</sub> =6	R <sub>7</sub> =4 R <sub>5</sub> =6 > 4 => R <sub>5</sub> =4 R <sub>2</sub> =∞ > 6 => R <sub>2</sub> =6	R <sub>1</sub> =∞ > 8 => R <sub>1</sub> =8 R <sub>2</sub> =6 < 9 => R <sub>2</sub> =6 R <sub>5</sub> =4	R <sub>2</sub> =5 < 7 => R <sub>2</sub> =5	R <sub>1</sub> =8 > 6 => R <sub>1</sub> =6	
Выбираем	R <sub>8</sub> =0	R <sub>7</sub> =1	R <sub>6</sub> =2	R <sub>4</sub> =2	R <sub>3</sub> =4	R <sub>5</sub> =4	R <sub>2</sub> =6	Стоп. Вершин нет

**если маршрут от А до В оптимален, то и оптимальный маршрут между двумя промежуточными вершинами принадлежит маршруту от А до В**



# Алгоритм Дейкстры

*Вопросы:*

- Сколько времени работает этот алгоритм?*
- Что происходит когда изменяется стоимость линии или когда маршрутизатор/линия выходят из строя?*

# Сложность алгоритма Дейкстры



Вариант реализации алгоритма Дейкстры	Насыщенный граф $m = O(n^2)$	Разреженный граф $m = O(n)$
<b>Вариант 1</b> $D[i]$ – это массив (поиск за время $O(n)$ )	$T = O(n^2 + m) =$ $O(n^2)$	$T = O(n^2 + m) =$ $O(n^2)$
<b>Вариант 2</b> $D[i]$ хранятся в бинарной куче	$T = O(n \log n + m \log n)$ $= O(n^2 \log n)$	$T = O(n \log n + m \log n)$ $= O(n \log n)$
<b>Вариант 3</b> $D[i]$ хранятся в Фибоначчиевой куче	$T = O(m + n \log n)$ $= O(n^2)$	$T = O(n + n \log n)$ $= O(n \log n)$

Граф разреженный:  $m = |A| \ll |V|^2 = n^2$



# Постановки задач о кратчайшем пути

- **Задача о кратчайшем пути между парой вершин (single-pair shortest path problem)**  
Требуется найти кратчайший путь из заданной вершины  $s$  в заданную вершину  $d$
- **Задача о кратчайших путях из заданной вершины во все (single-source shortest path problem)**  
Найти кратчайшие пути из заданной вершины  $s$  во все
- **Задача о кратчайшем пути в заданный пункт назначения (single-destination shortest path problem)**  
Требуется найти кратчайшие пути в заданную вершину  $v$  из всех вершин графа
- **Задача о кратчайшем пути между всеми парами вершин (all-pairs shortest path problem)**  
Требуется найти кратчайший путь из каждой вершины  $u$  в каждую вершину  $v$



# Алгоритмы поиска кратчайшего пути в графе

Алгоритм	Применение
Алгоритм Дейкстры	Находит кратчайший путь от одной из вершин графа до всех остальных. Алгоритм работает только для графов без ребер отрицательного веса ( $w_{ij} \geq 0$ )
Алгоритм Беллмана-Форда	Находит кратчайшие пути от одной вершины графа до всех остальных во взвешенном графе. Вес ребер может быть отрицательным
Алгоритм поиска $A^*$ (A star)	Находит путь с наименьшей стоимостью от одной вершины к другой, используя алгоритм поиска по первому наилучшему совпадению на графе
Алгоритм Флойда-Уоршелла	Находит кратчайшие пути между всеми вершинами взвешенного ориентированного графа
Алгоритм Джонсона	Находит кратчайшие пути между всеми парами вершин взвешенного ориентированного графа (должны отсутствовать циклы с отрицательным весом)
Алгоритм Ли (волновой алгоритм)	Находит путь между вершинами $s$ и $t$ графа, содержащий минимальное количество промежуточных вершин (трассировки электрических соединений на кристаллах микросхем и на печатных платах)
Алгоритмы Viterbi, Cherkassky, ...	





# Алгоритм Дейкстры на практике

- *Алгоритм Дейкстры - это пример алгоритма по состоянию канала*
  - *состояние линии знает каждый маршрутизатор*
  - *каждый маршрутизатор строит соединяющее дерево минимальной стоимости до каждого другого маршрутизатора*
- *Этот алгоритм является основой OSPF (Open Shortest Path First) - широко используемого протокола маршрутизации (том 2 стр.82-86)*



# Маршрутизация в Интернет

(том 2 стр.82-88)

Введение в компьютерные сети

чл.-корр. РАН. Смелянский Р.А.

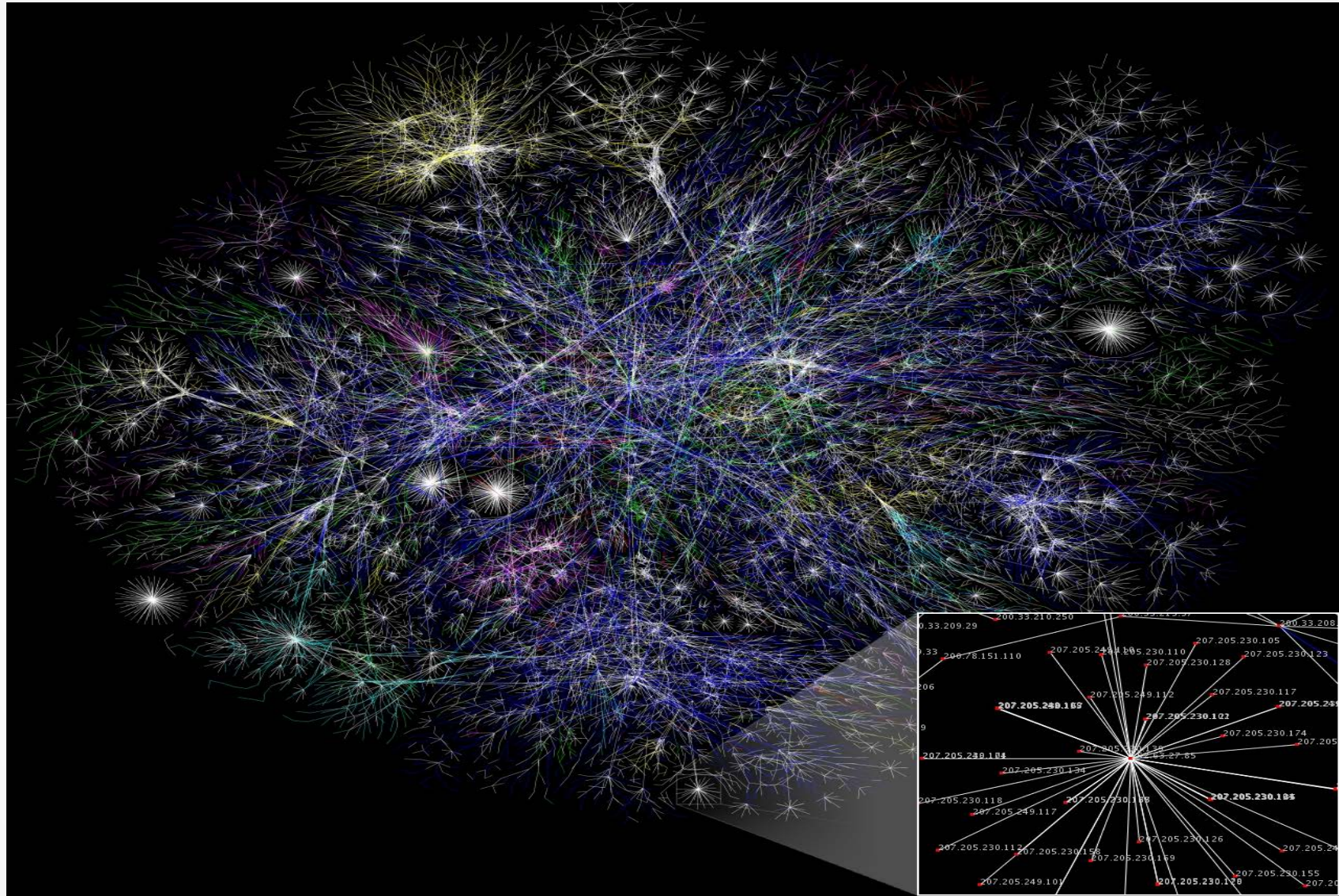
Кафедра АСВК

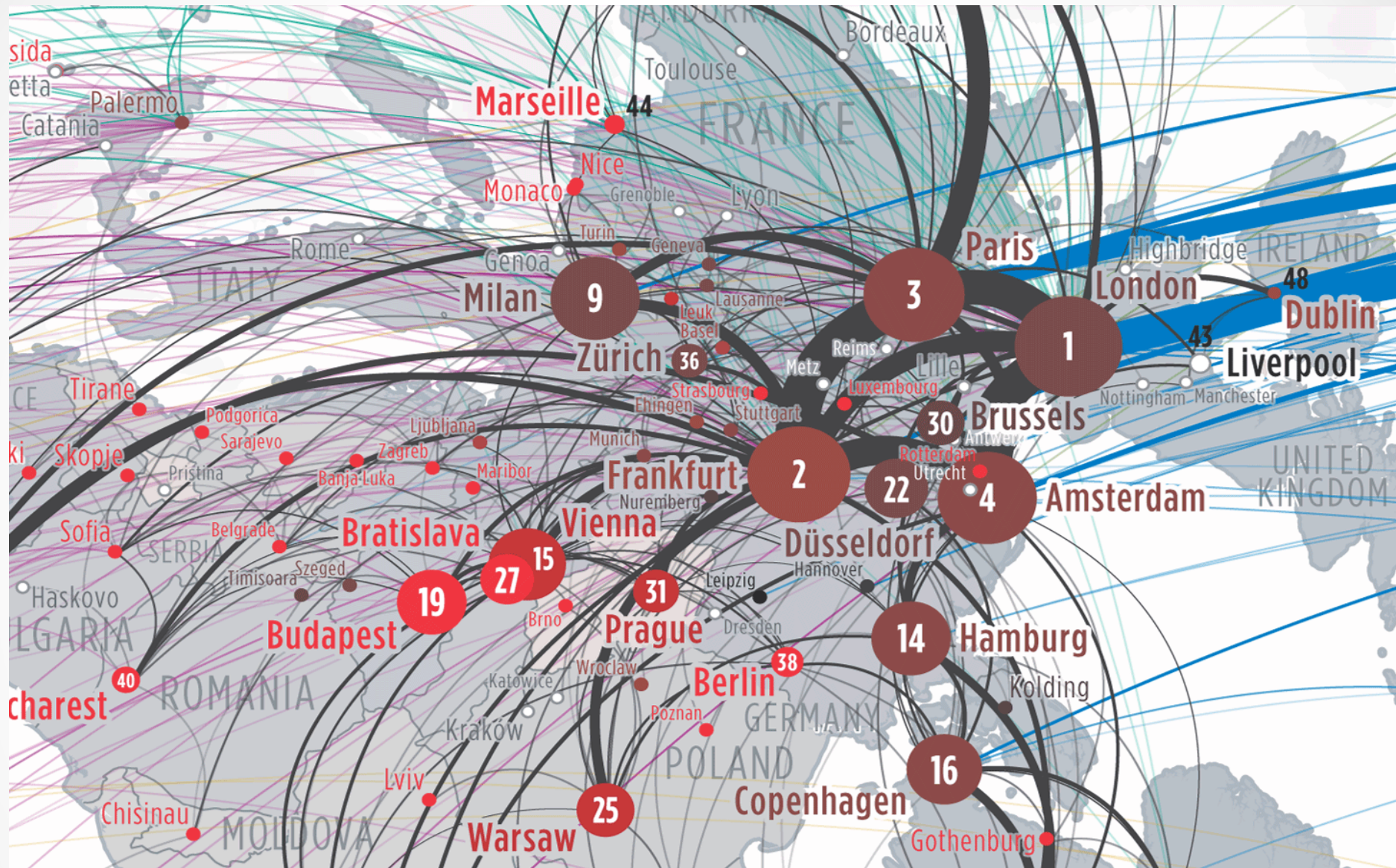
ф-т ВМК МГУ



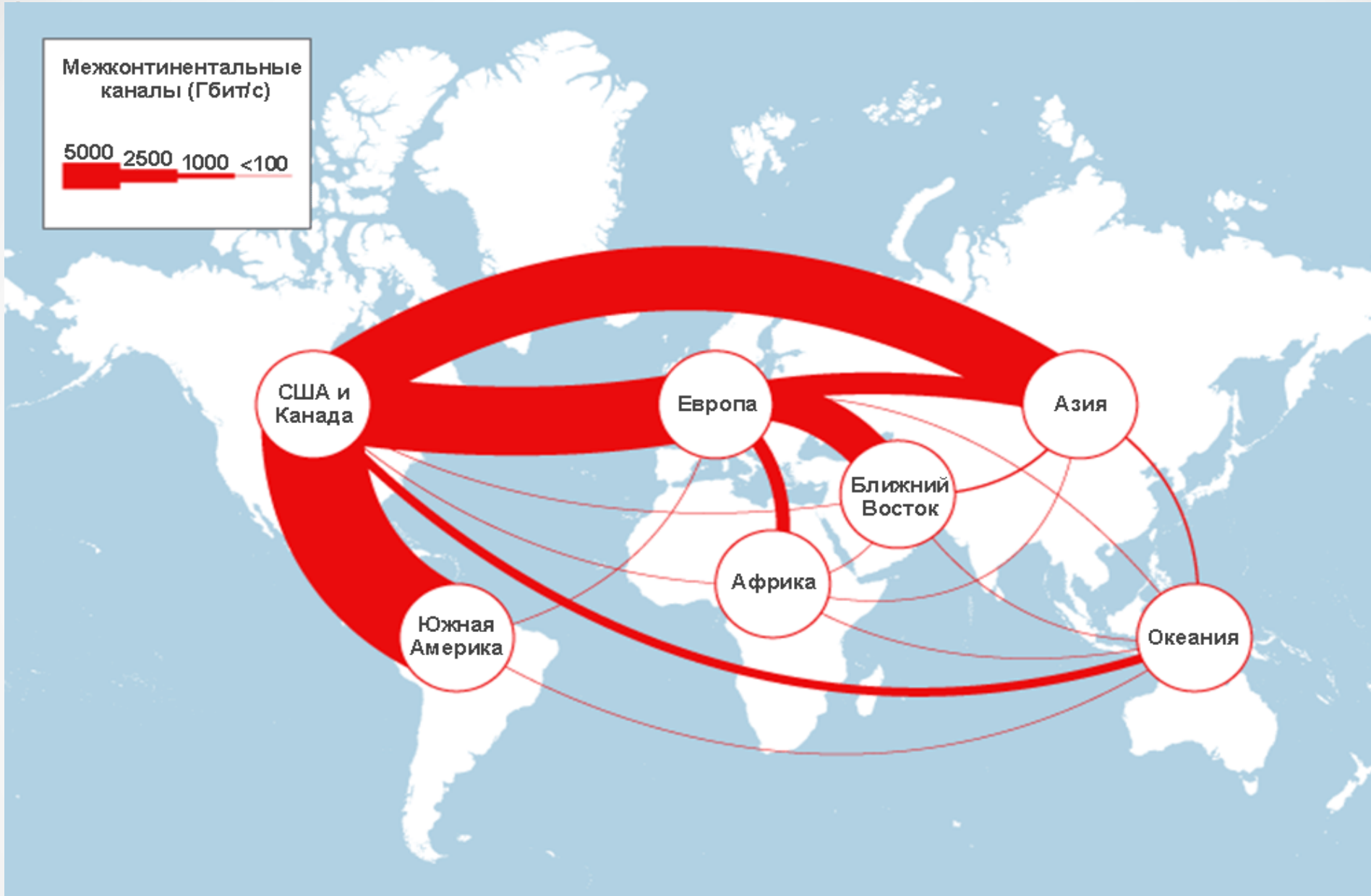
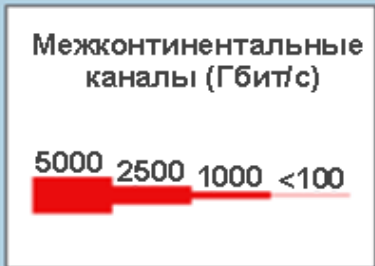


# Как быть с такой сетью ?





Если магистральные каналы связи сравнить с кровеносной системой современной цивилизации, то Европа — её сердце.





# Автономные Системы

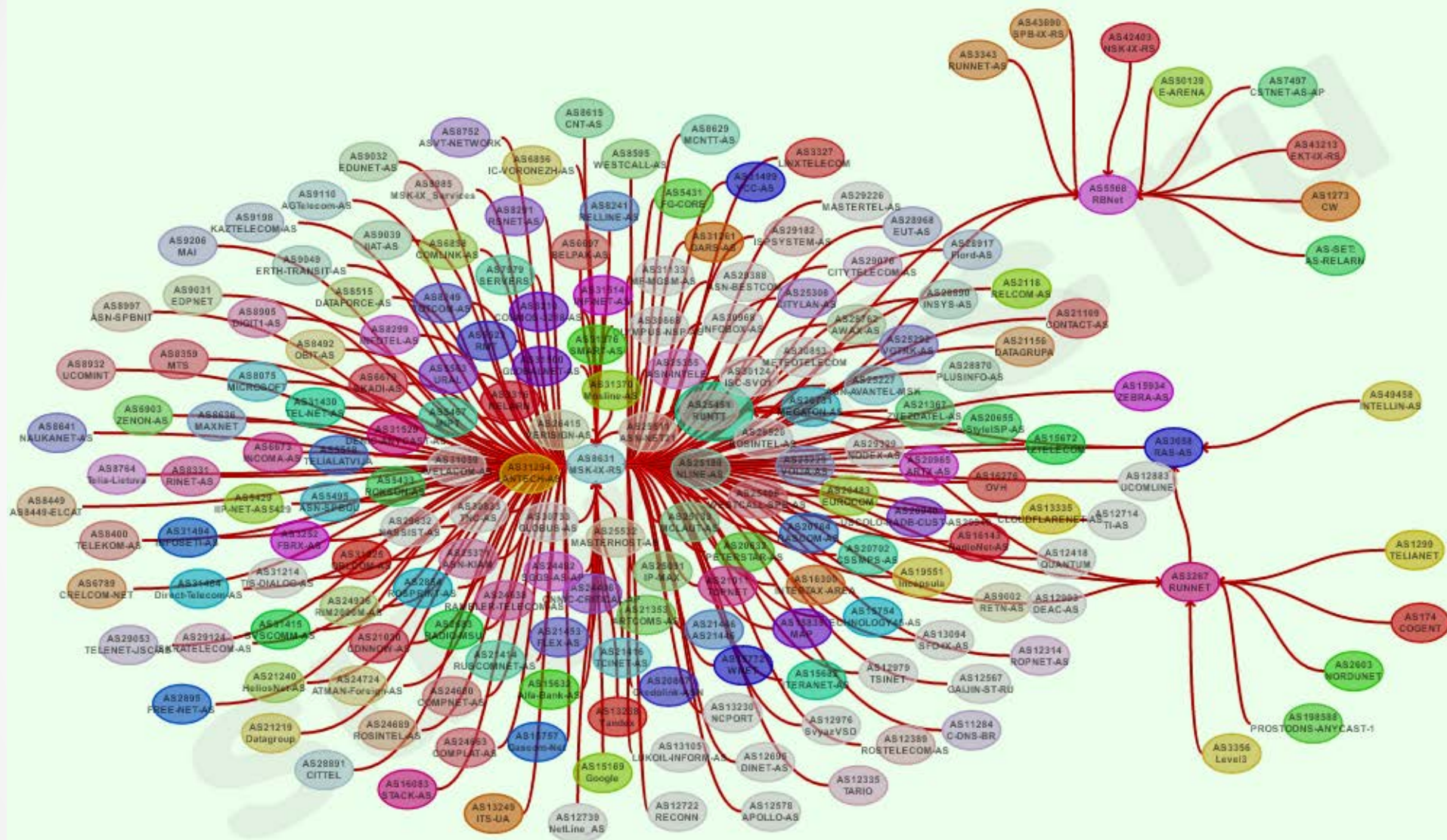
## Единица иерархии в Интернет

- *Внутри АС ее владелец решает как маршрутизировать потоки данных*
- *Между АС должен использоваться протокол BGP-4 (Border Gateway Protocol v4 RFC 1771)*

## Как найти номер АС?

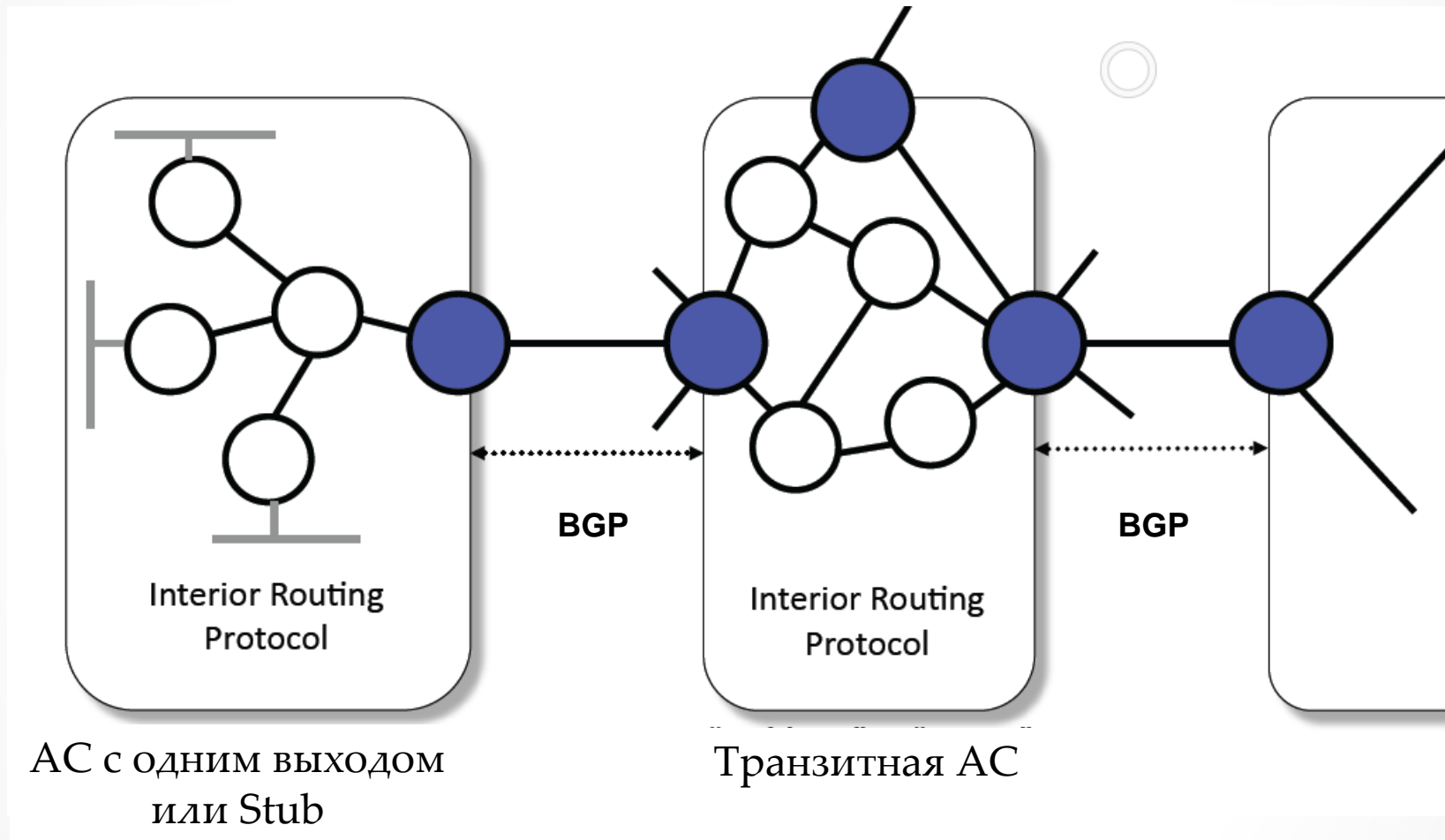
<http://whatismyipaddress.com/ip-lookup>

# Связность автономных систем





# Иерархическая структура Интернет



IRP – RIP (практически не используется), OSPF, IS-IS, IBGP





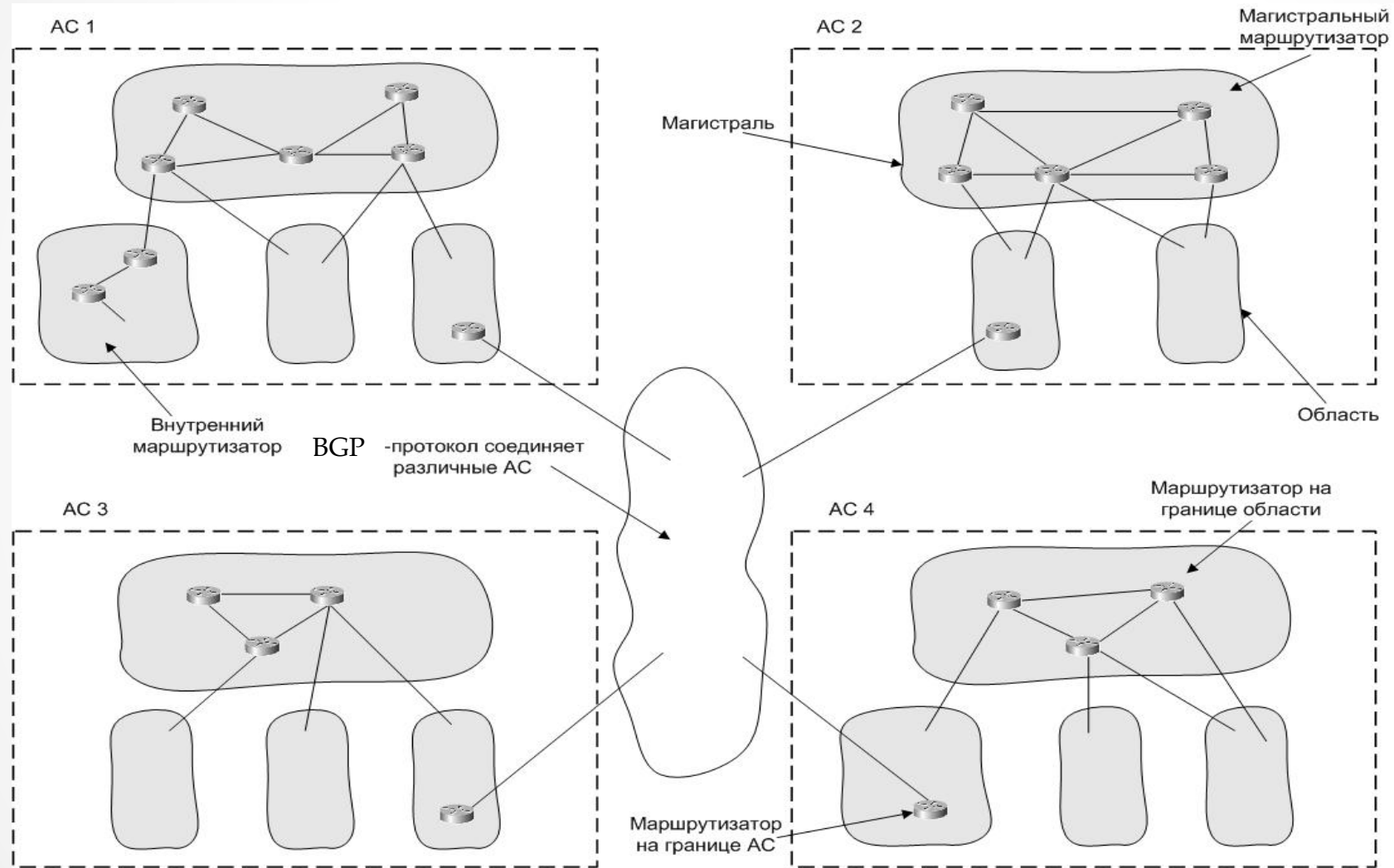
# Маршрутизация в и между AS

*В AS может быть несколько точек входа и выходы ( несколько BGP шлюзов). Если у AS несколько BGP шлюзов, то эти шлюзы взаимодействуют через IBGP протокол. Все остальные маршрутизаторы одной и той же AS взаимодействуют по OSPF протоколу.*

- Каждый маршрутизатор знает все префиксы внутри AS*
- Пакеты для других AS пересылаются на ближайший IBGP шлюз-маршрутизатор*
- EBGP маршрутизатор - пограничный шлюз для других AS*



# AS, магистрали и области OSPF





# Протокол OSPF

- *АС можно разбивать на области*
- *OSPF (RFC 2328)*
- *изменения состояний линии рассылаются лавиной по необходимости*
- *каждый маршрутизатор использует алгоритм Дейкстры*
- *изменения аутентифицируются*



# OSPF протокол

1. Маршрутизаторы обмениваются маленькими HELLO-пакетами
2. Обменявшись пакетами, они устанавливают соседские отношения, добавляя каждый друг друга в свою локальную таблицу соседей
3. Маршрутизаторы собирают состояния всех своих линий (связей с соседями), включающие в себя id маршрутизатора, id соседа, сеть и префикс между ними, тип сети, стоимость линии (метрику) и формируют пакет, называемый LSA (Link State Advertisement).
4. Маршрутизатор рассылает LSA своим соседям, те распространяют LSA дальше.
5. Каждый маршрутизатор, получивший LSA добавляет в свою локальную табличку LSDB (Link State Database) информацию из LSA.
6. В LSDB скапливается информация, обо всех парах соединённых в сети маршрутизаторов, т.е. каждая строка таблицы — это информация вида: «Маршрутизатор А имеет соединение со своим соседом маршрутизатором В, между ними сеть такая-то с такими-то характеристиками».
7. После обмена LSA, каждый маршрутизатор знает про все линии, на основании пар строится полная карта сети, включающая все маршрутизаторы и все связи между ними.
8. Среди всех своих соседей маршрутизатор выбирает смежного соседа (designated), через которого он будет рассылать LSA сообщения в другие области
9. На основании этой карты каждый маршрутизатор индивидуально ищет кратчайшие по метрике маршруты во все сети из других областей и добавляет их в таблицу маршрутизации.

# OSPF Типы маршрутизаторов



- **Внутренний маршрутизатор (internal router)** — маршрутизатор, все интерфейсы которого принадлежат одной области. У таких маршрутизаторов одинаковая база данных состояния каналов.
- **Пограничный маршрутизатор (area border router, ABR)** — соединяет одну или больше областей с магистральной областью и выполняет функции шлюза для транзитного трафика.
- **Магистральный маршрутизатор (backbone router)** — маршрутизатор у которого всегда хотя бы один интерфейс принадлежит магистральной области.
- **Пограничный маршрутизатор автономной системы (AS boundary router, ASBR)** — это маршрутизатор, один из портов которого находится в домене OSPF протокола, а другой в домене любого из внутренних шлюзовых протоколов (например RIP или IGRP).

## OSPF протокола (продолжение)



- 4) Маршрутизаторы подключены через общую область/сеть с множественным доступом (транзитная область)
- 5) Отношение соседства потребовало бы  $n^2$  LSA
- 6) Для того чтобы преодолеть эту проблему в OSPF существует механизм выбора **Designated Router (DR)** и **Backup Designated Router (BDR)**. DR и BDR - это роли маршрутизаторов.

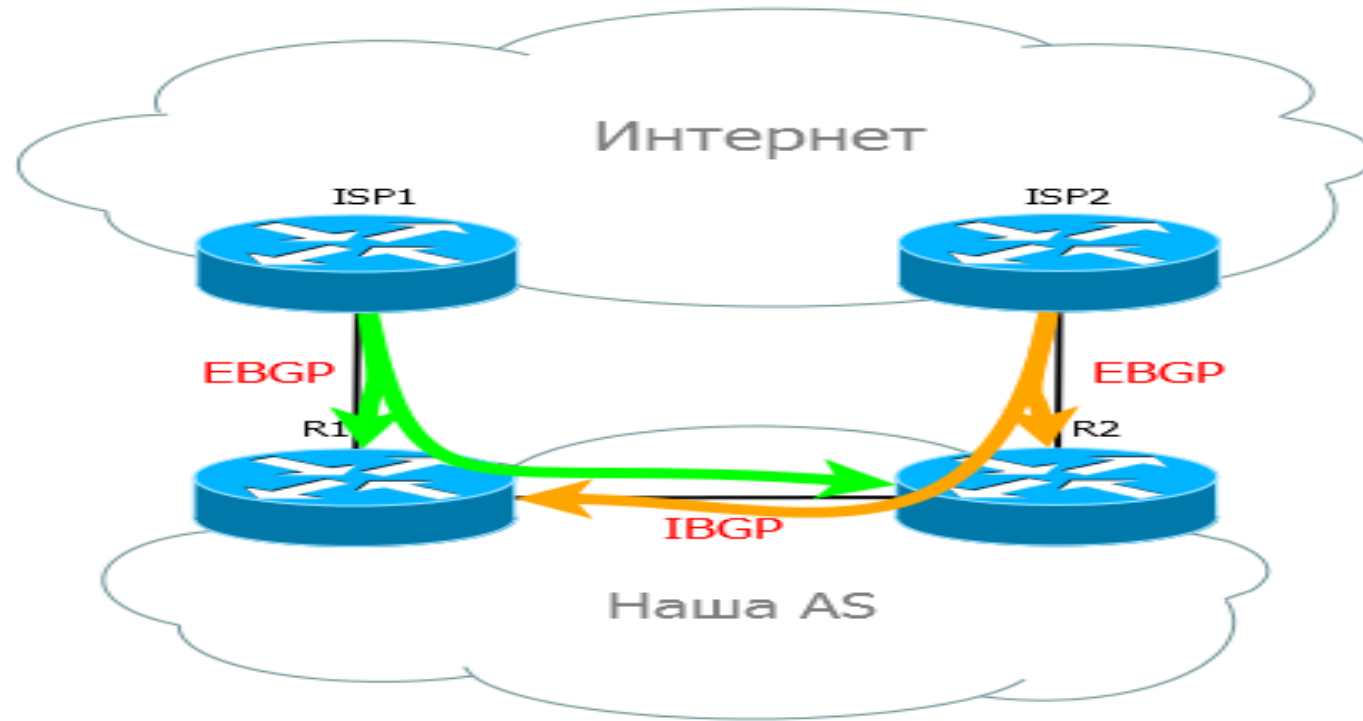


# Маршрутизация в AS через несколько точек выхода

- *Используется в транзитных AS и сетях международных компаний с разветвленной сетью офисов*
- *Каждому внутреннему маршрутизатору должно быть сообщено какую точку выхода он должен использовать для определенного префикса точки назначения*
- *Таблица маршрутизации существенно разрастается*
- *Два подхода:*
  - *«горячая картошка» - переслать ближайшему выходу*
  - *выбрать выход ближе всего к токе назначения*



# Внутренний межщлюзовой протокол маршрутизации - IBGP

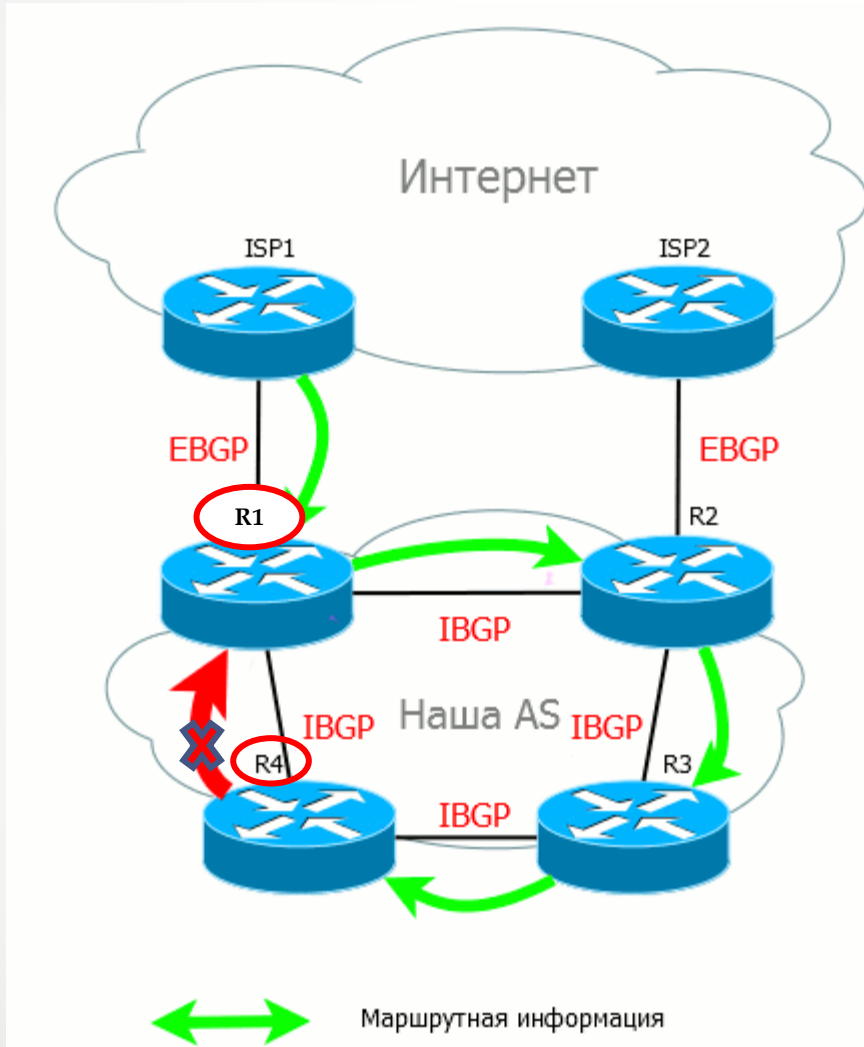


Маршрутная информация





# Внутренний межшлюзовой протокол маршрутизации - IBGP



- В случае полностью связной топологии правила *Split Horizon* исключают ситуацию возникновения циклов при появлении новых маршрутизаторов (R4).
- R1 получил объявление от ISP1, немедленно передает его всем своим соседям: R2, R3, R4. Но R4 не передает информацию R1, т.к. она получена от R1. Потому все маршрутизаторы BGP имеют актуальную информацию, и петли исключены.

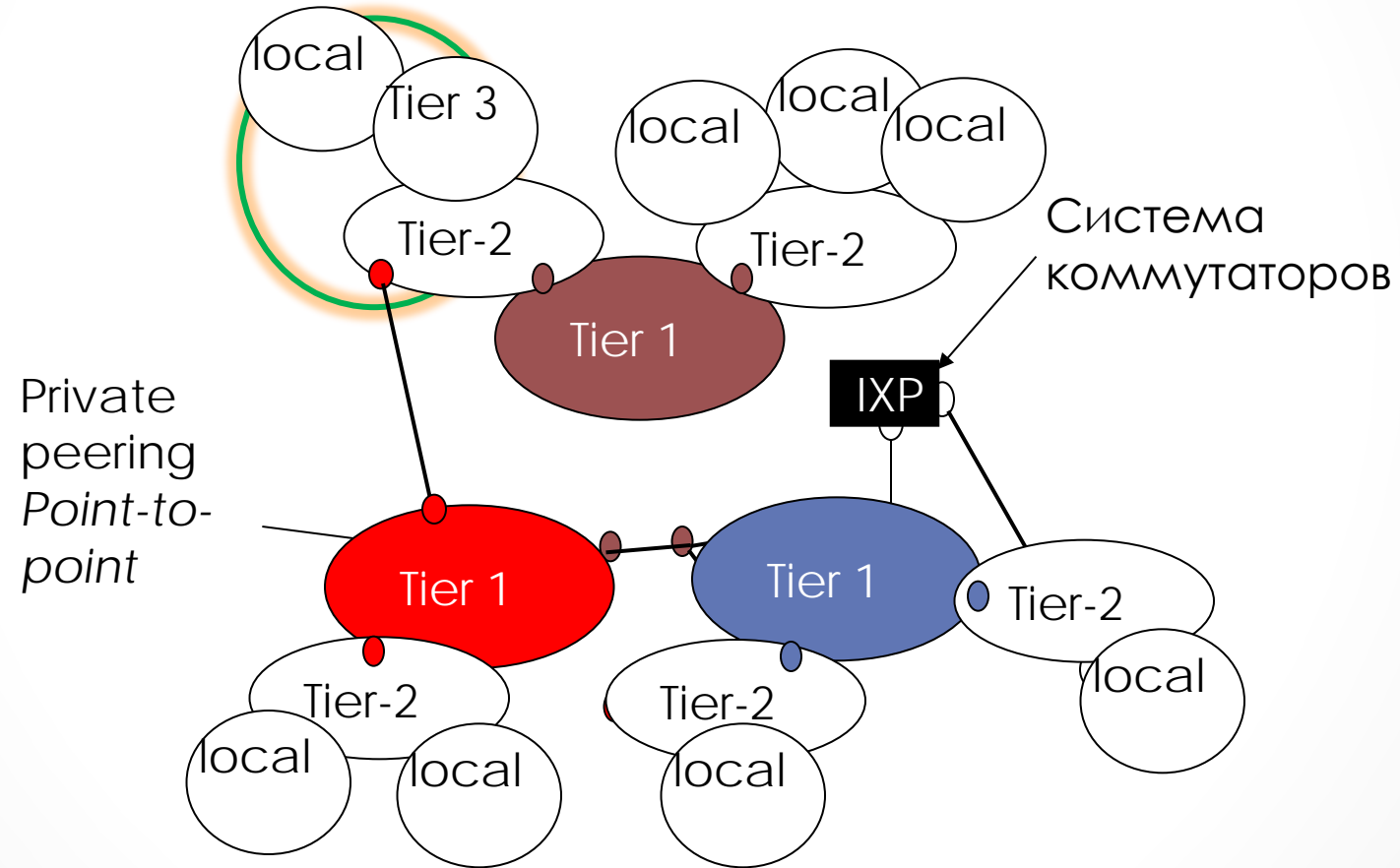


# Протокол внешней маршрутизации

- Все АС взаимодействуют, используя протокол BGP-4
- BGP-4 был разработан чтобы решить следующие проблемы:
  - **Топология:** Интернет плохо структурированная смесь разнообразных АС
  - **Автономия АС:** каждая АС по-своему определяет стоимость линии, поэтому невозможно построить путь с наименьшей стоимостью
  - **Доверие:** некоторые АС не могут доверять тем маршрутам, которые предлагают другие АС (два конкурирующих провайдера, защита конф идентичности через территорию неприятеля)
  - **Политика:** разные АС преследуют разные цели (мин. число скачков vs предпочтение одного провайдера перед другими)

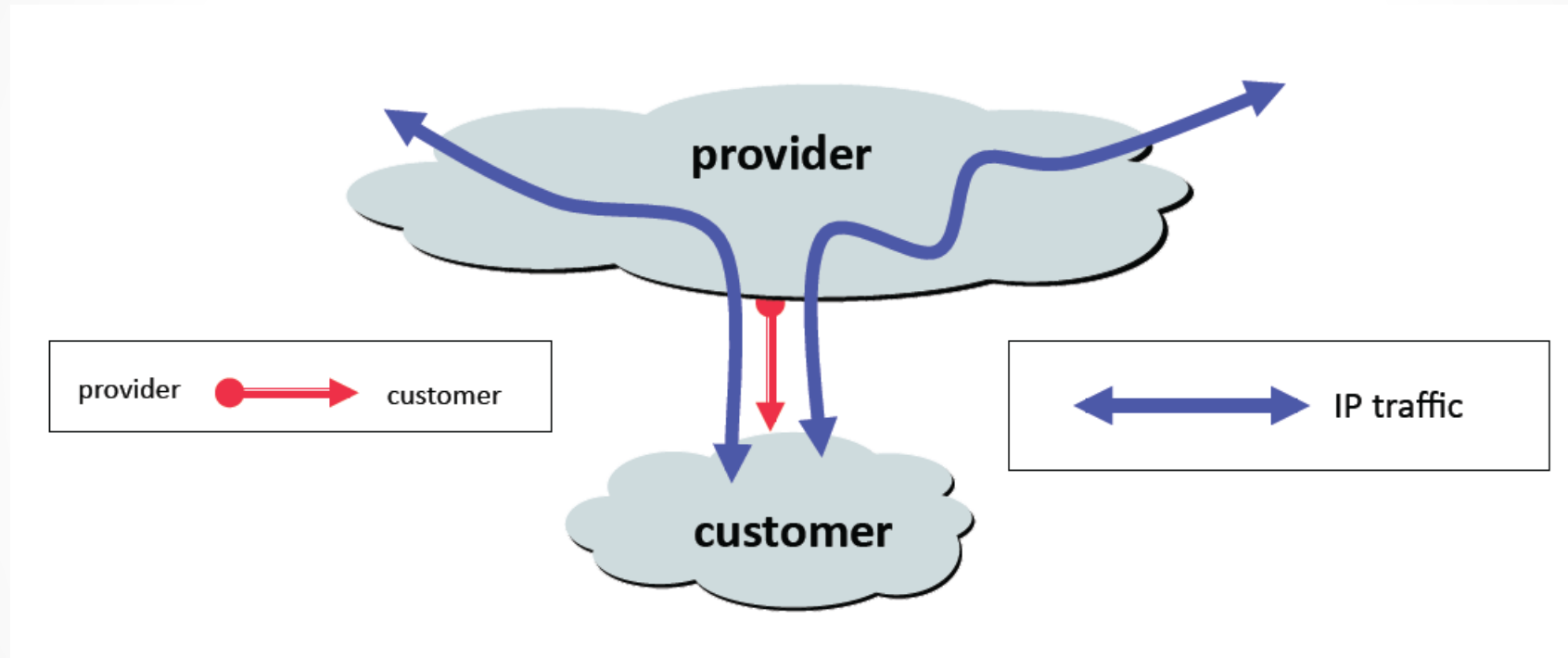


# Иерархия ISP - Структура Интернет





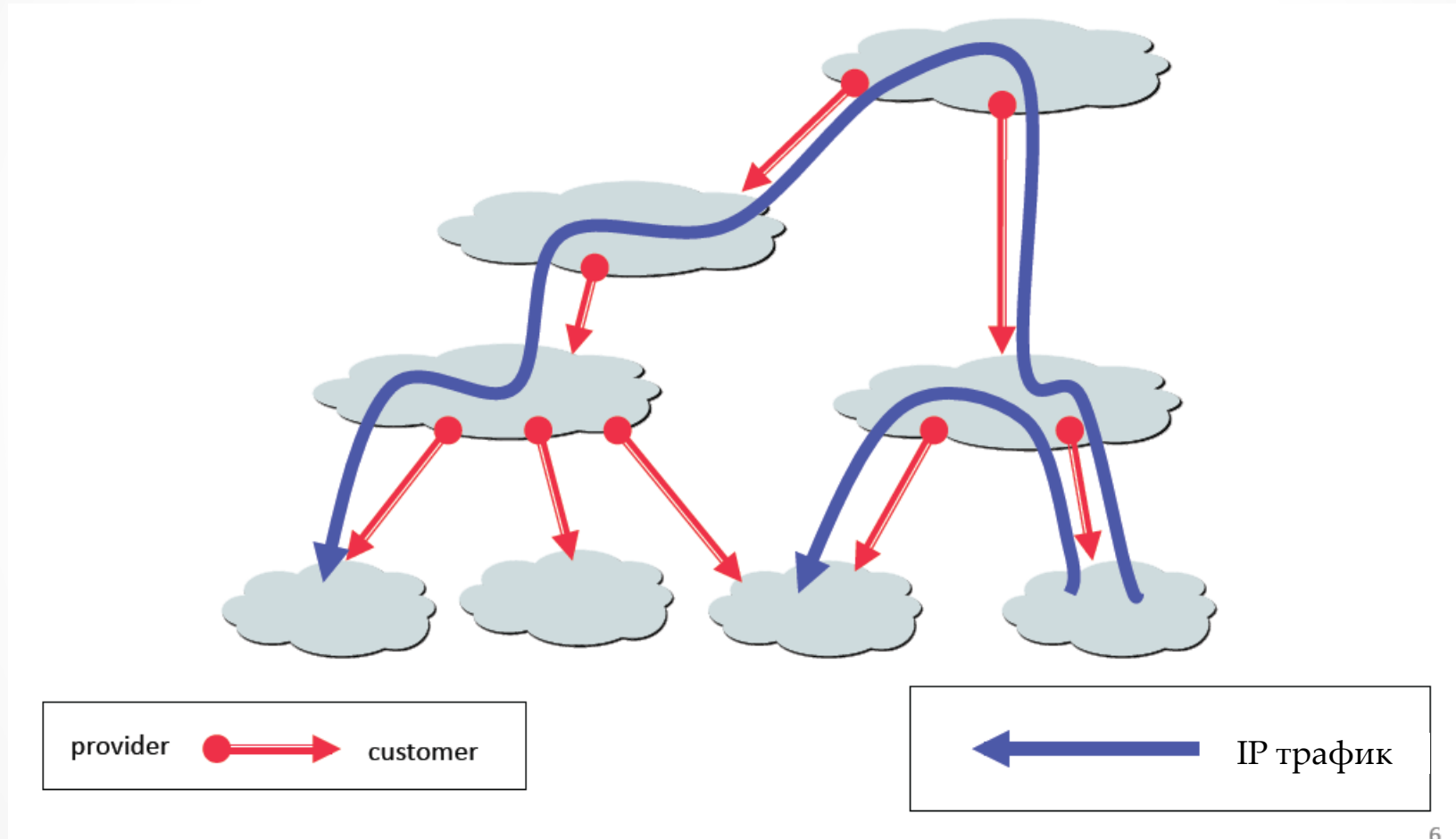
# Заказчики и провайдеры



*Заказчики платят провайдеру за свои пакеты*



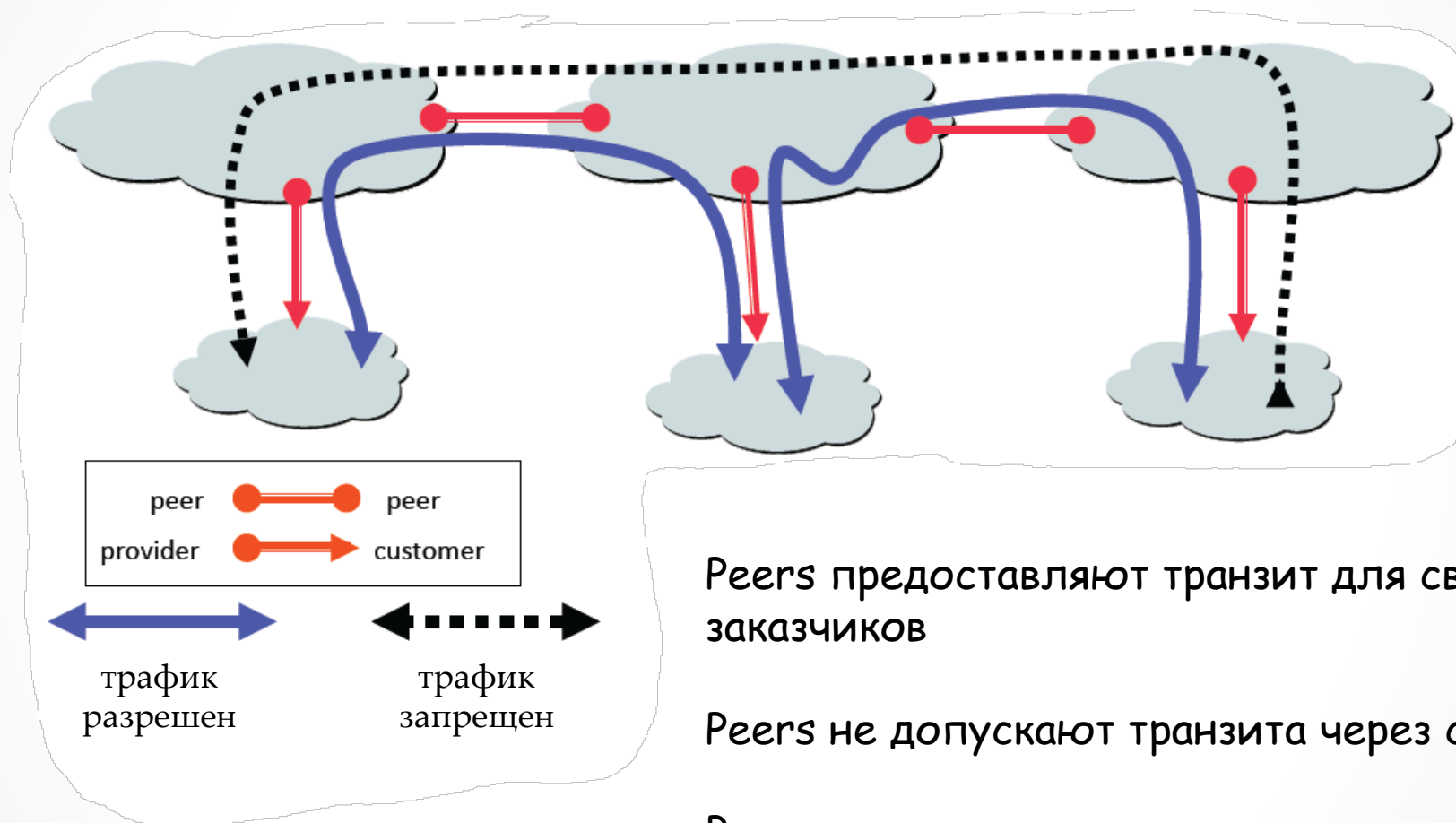
# Иерархия заказчиков и провайдеров



6



# Отношение Peering



Peers предоставляют транзит для своих важных заказчиков

Peers не допускают транзита через себя другим peers

Peers не ведут, как правило, взаиморасчетов



## Итог

- Интернет состоит из множества независимо управляемых АС
- Каждая АС использует свой внутренний протокол маршрутизации на уровне областей и IGP протокол между областями
- Оконечные АС используют простую маршрутизацию по умолчанию
- Транзитные АС должны сами определять какой выход использовать
- Для взаимодействия АС должны использовать BGP-4 протокол



# Маршрутизация между AS - BGP протокол

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.А.

Кафедра АСВК

ф-т ВМК МГУ





# ОСНОВЫ Border Gateway Protocol (BGP-4)



- BGP использует «вектор пути»
- Каждый BGP маршрутизатор рассылает список путей (путь - список AS)
  - AS\_PATH
  - К сети 171.64/16 можно пройти по пути {AS7,AS52,AS13}
- Наличие цикла в маршруте определяется локально и такие маршруты игнорируются
- Из множества доступных маршрутов выбирается тот, который наиболее всего соответствует политике AS
- Если маршрутизатор/линии вышли из строя, то маршрут изымается из списка



# BGP packets

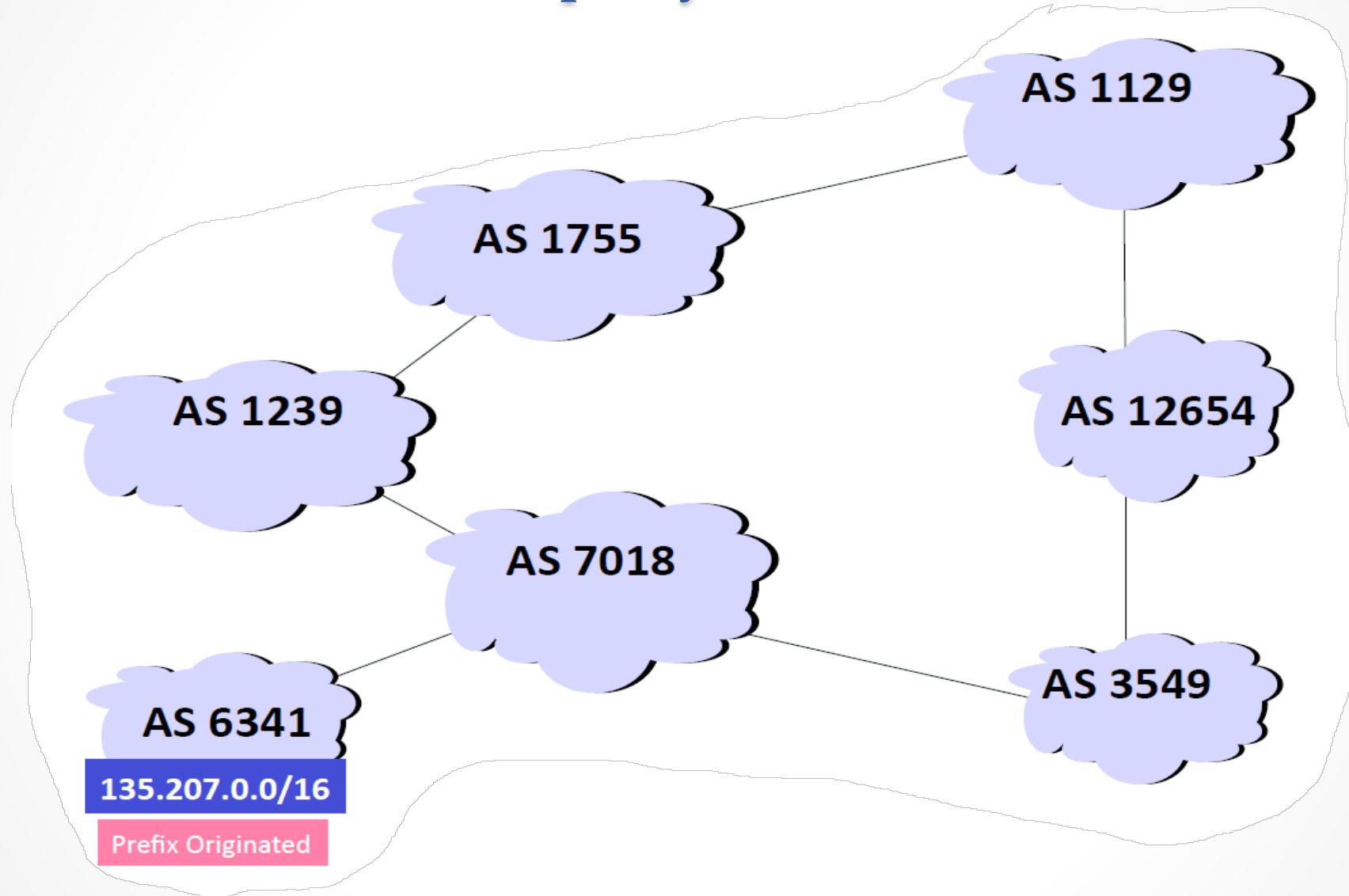
(<http://telenetwork.narod.ru/books/cisco/Mamaev/telecomtech/bgp.html>)

**Open** - BGP session setup (TCP port 179)

- **Keep Alive** - Health check at regular intervals
- **Notification** - Closing peering session
- **Update** - Announcing a new route or withdrawing a previously announced route
- **BGP объявление = префикс + атрибуты маршрута**
- **Path attributes**
  - Origin - who suggested this route
  - AS path - preferences, the rule for determining the output
  - next hop - IP address of BGP gateway of the next AS,
  - multi\_exit\_disc - how to select one of several gateways advertizing the rought
  - Local pref - prefixes of local speakers
  - Aggregate - combining several routes with a common prefix

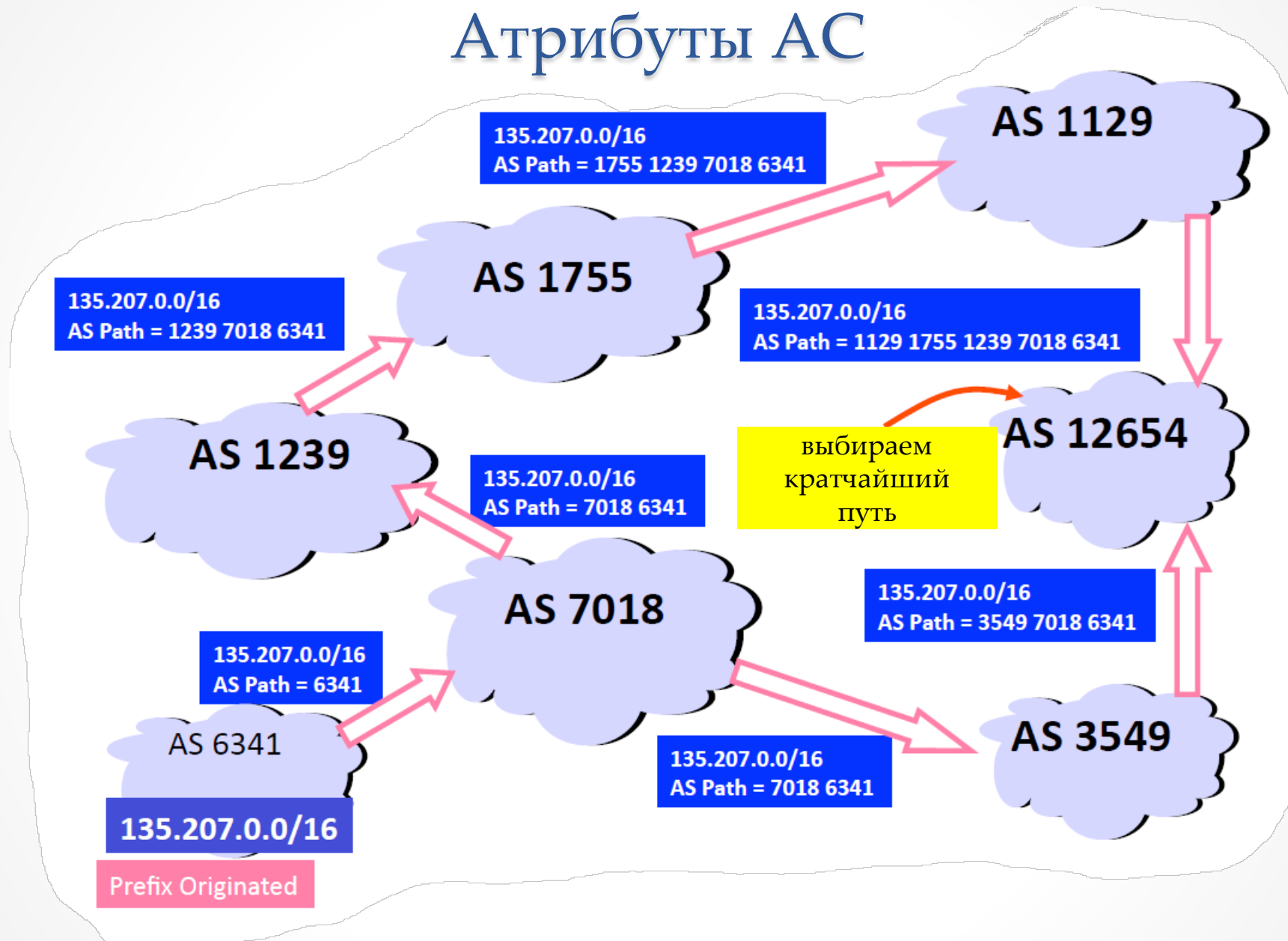


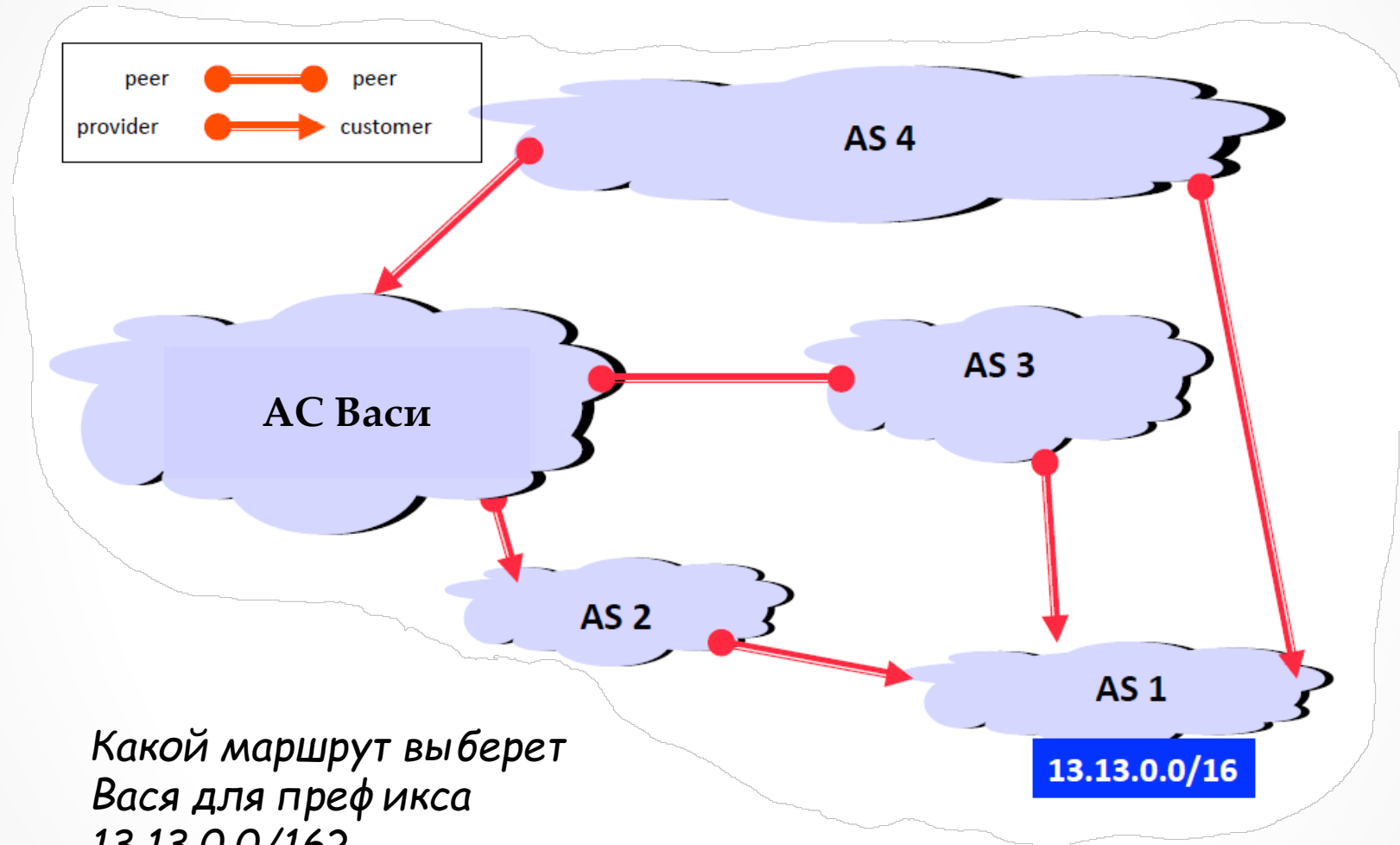
# Атрибуты АС



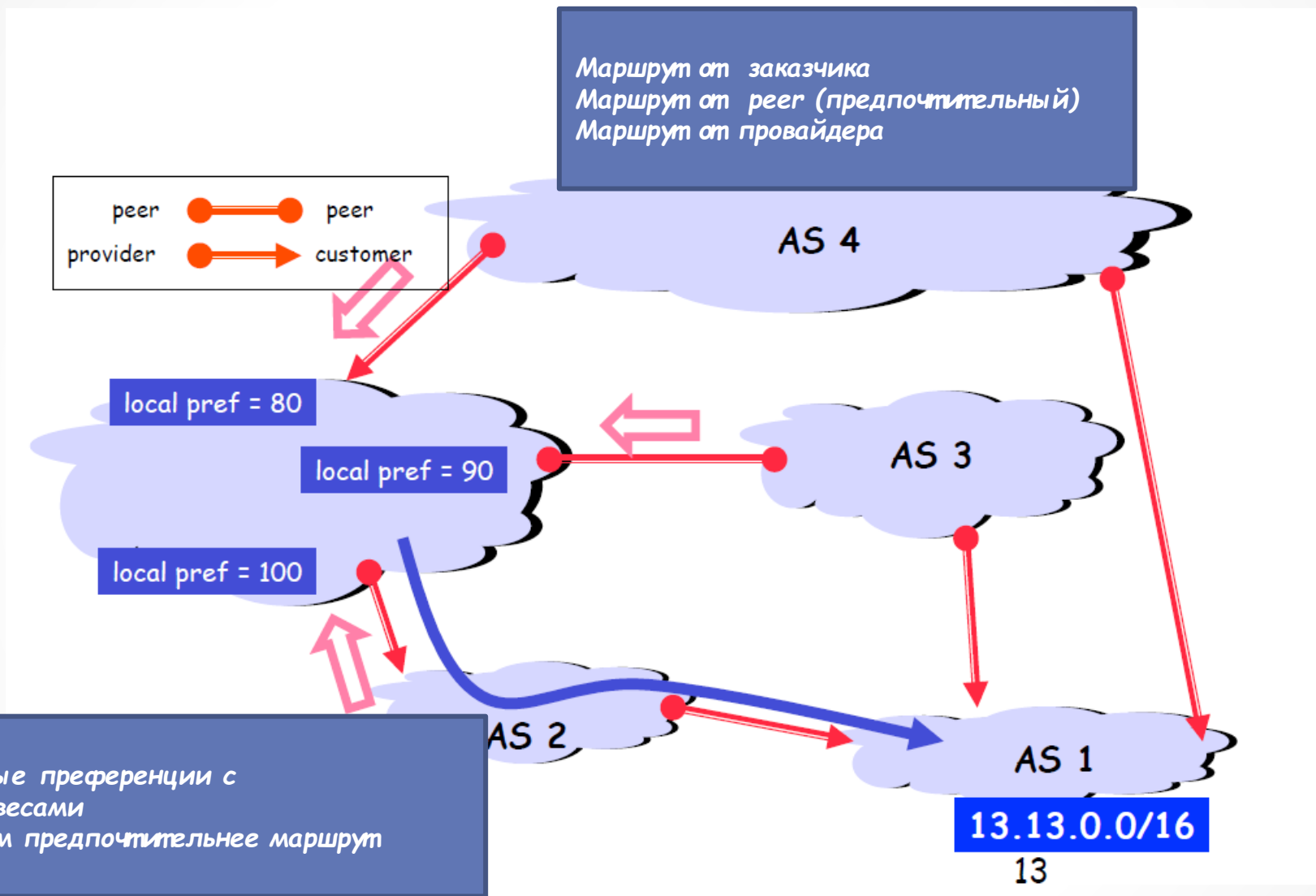


# Атрибуты АС

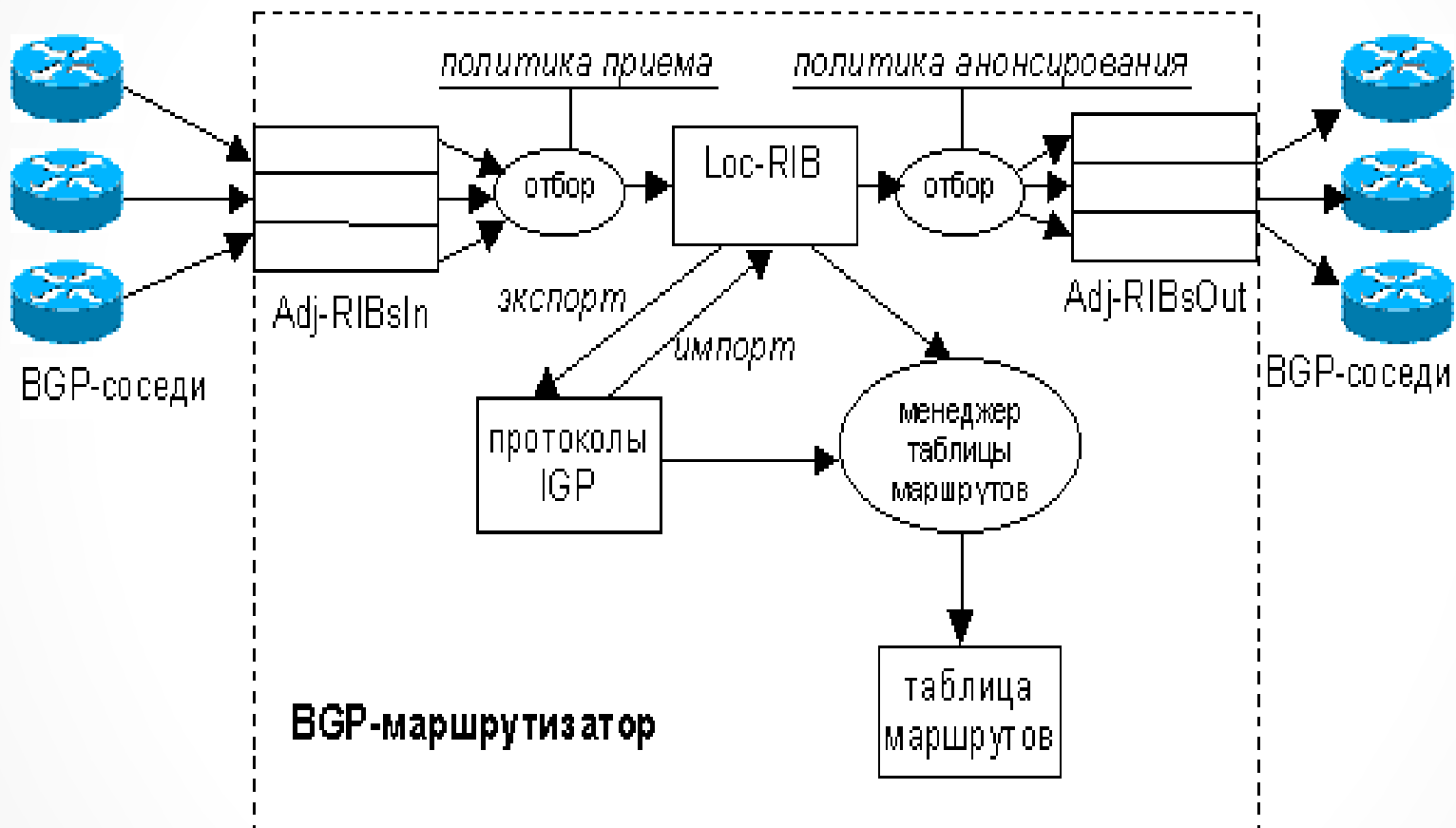




Какой маршрут выберет  
Вася для префикса  
13.13.0.0/16?



# Процедура выбора маршрута





# IBGP & EBGP различия

## 1) Устойчивость к образованию петель

- *EBGP* устраняет циклы с помощью *AS-Path*. Если в списке уже был номер локальной *AS*, этот маршрут отбрасывается.
- Когда маршрут объявляют внутри автономной системы, *AS-Path* не меняется. *IBGP* использует полносвязную сеть (*mesh*).
- В этом случае маршрут, полученный от соседа *IBGP*, не объявляется другим соседям *IBGP*. Это позволяет всем маршрутизаторам иметь все маршруты и при этом избегать петель.





## Итог

- *Все АС для взаимодействия в Интернете должны использовать BGP-4*
- *BGP-4 использует алгоритм маршрутизации по вектору пути, которые легко распознает циклы*
- *BGP-4 имеет сложный интерфейс, позволяющий каждой АС устанавливать свою локальную политику маршрутизации*
- *Каждая АС устанавливает свою политику для построения маршрутов, безопасности и локальных особенностей*



# Групповая маршрутизация (multicast routing)

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.А.

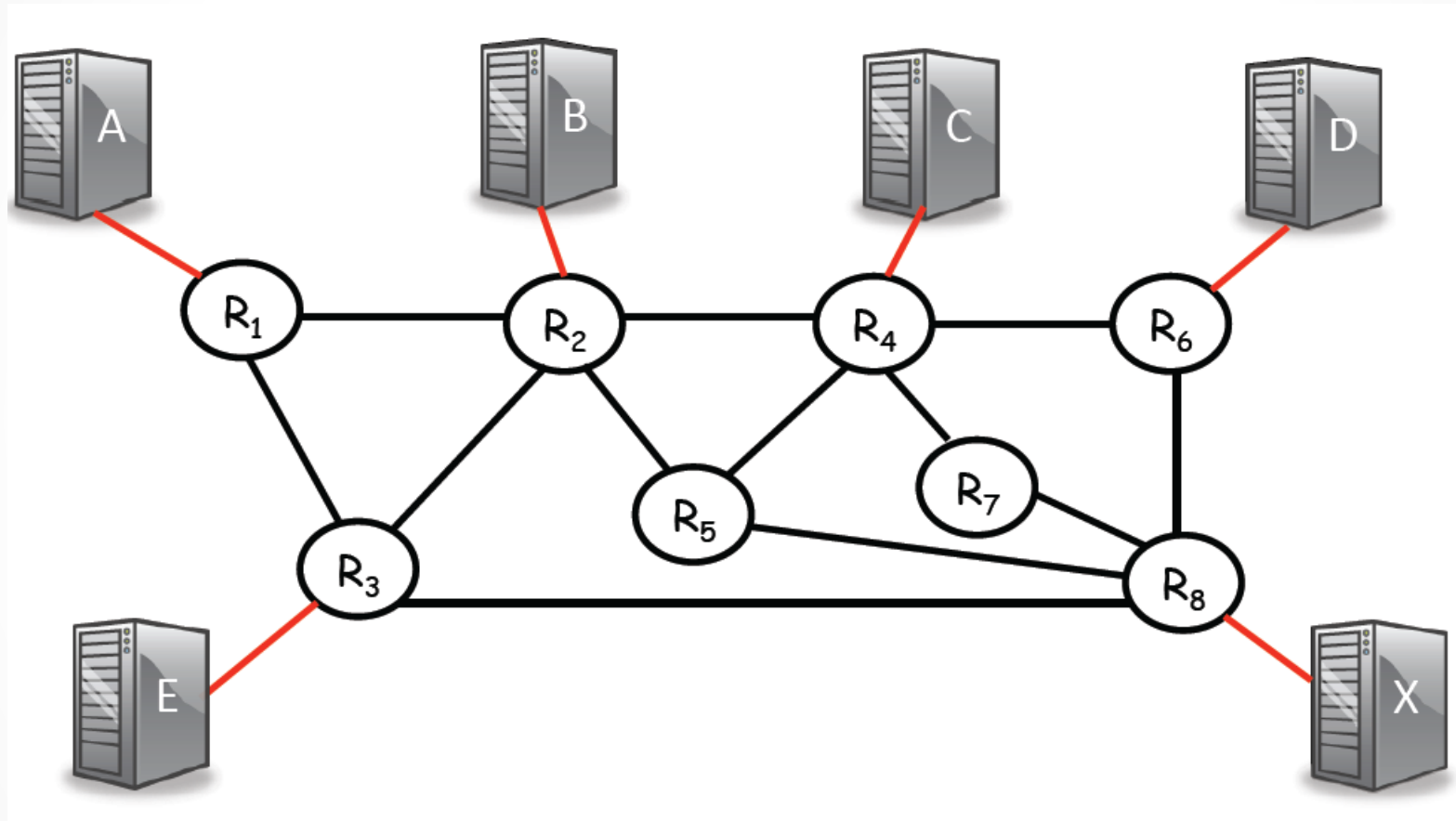
Кафедра АСВК

ф-т ВМК МГУ



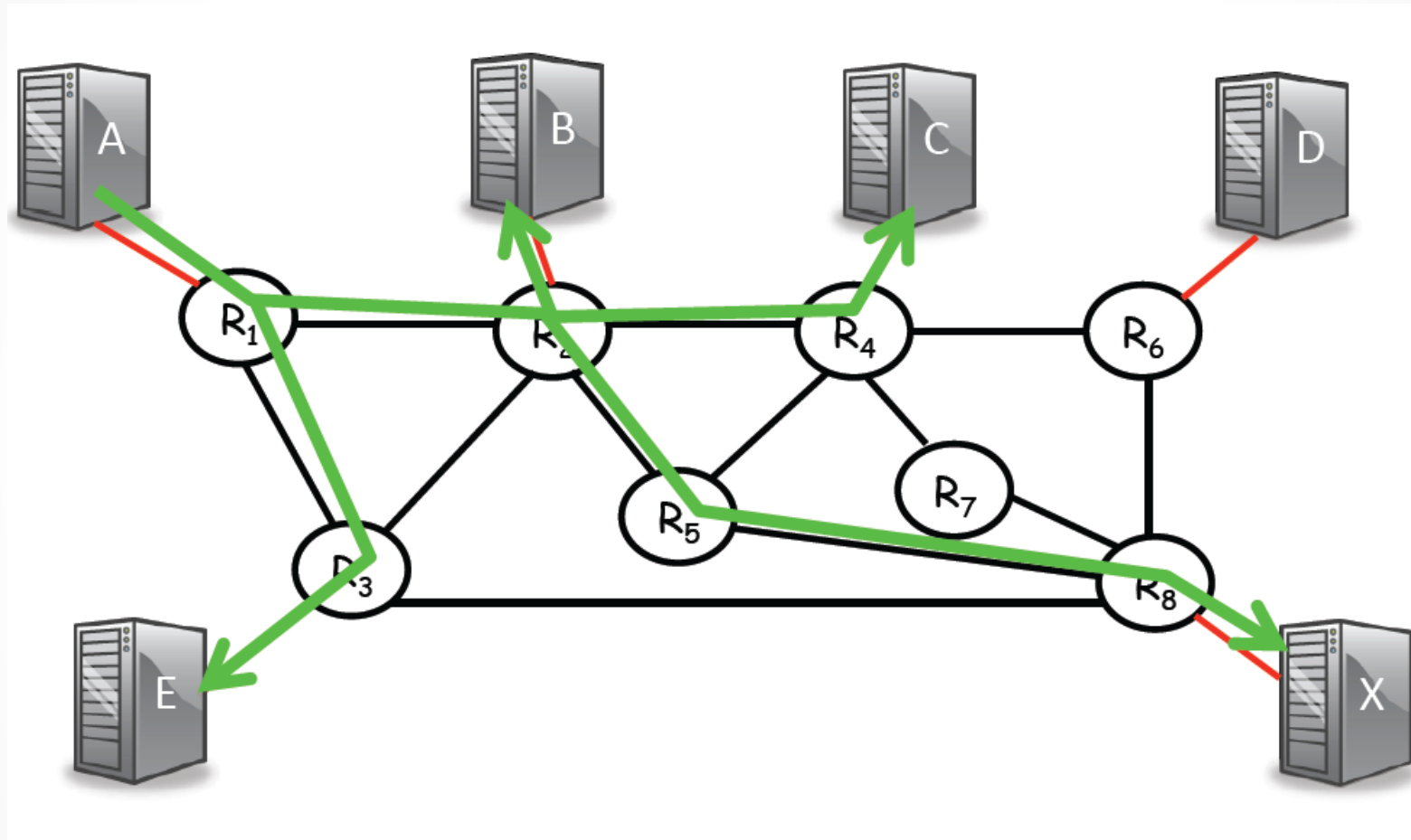


# Групповая маршрутизация



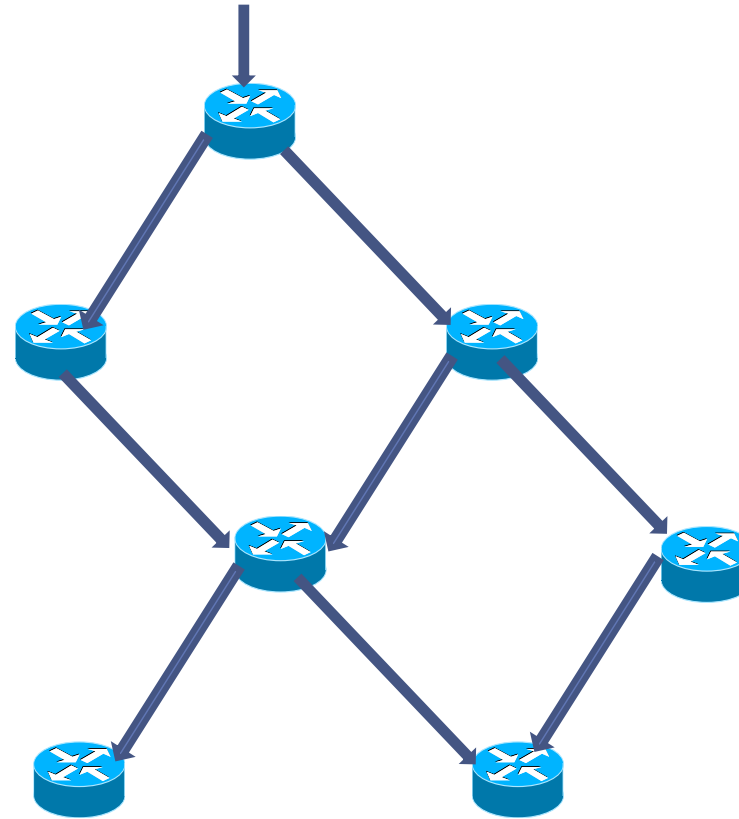


# Групповая маршрутизация





# Лавиной (Flooding)

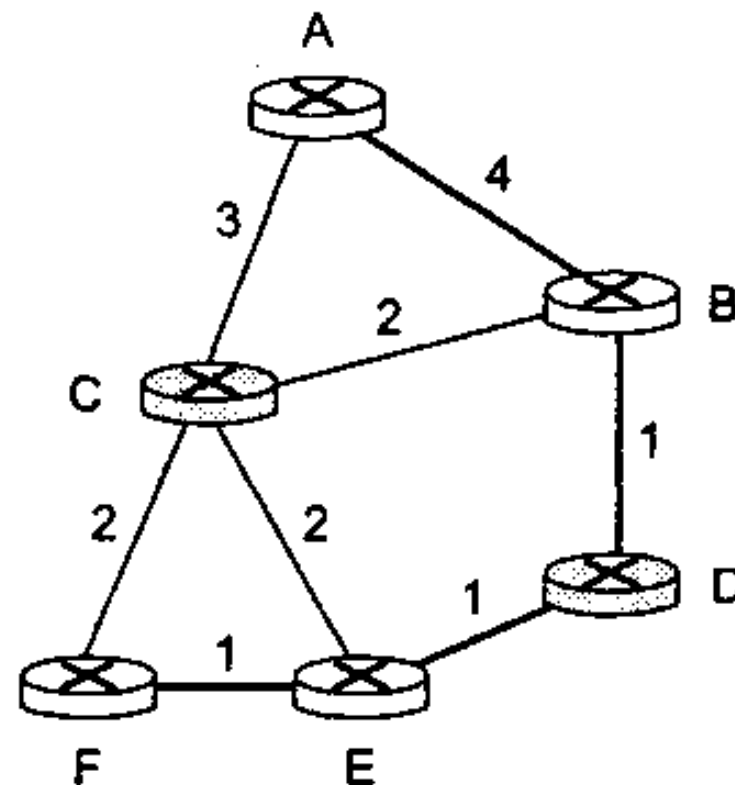
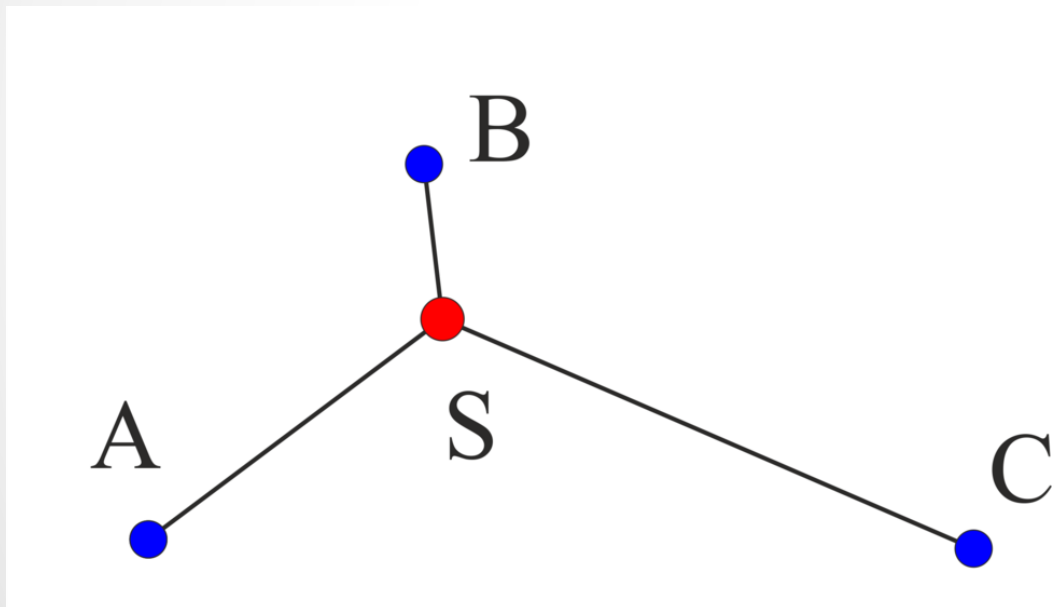




# Групповая маршрутизация

*Задача Штейнера о минимальном дереве состоит в поиске кратчайшей сети, соединяющей заданный конечный набор точек плоскости.*

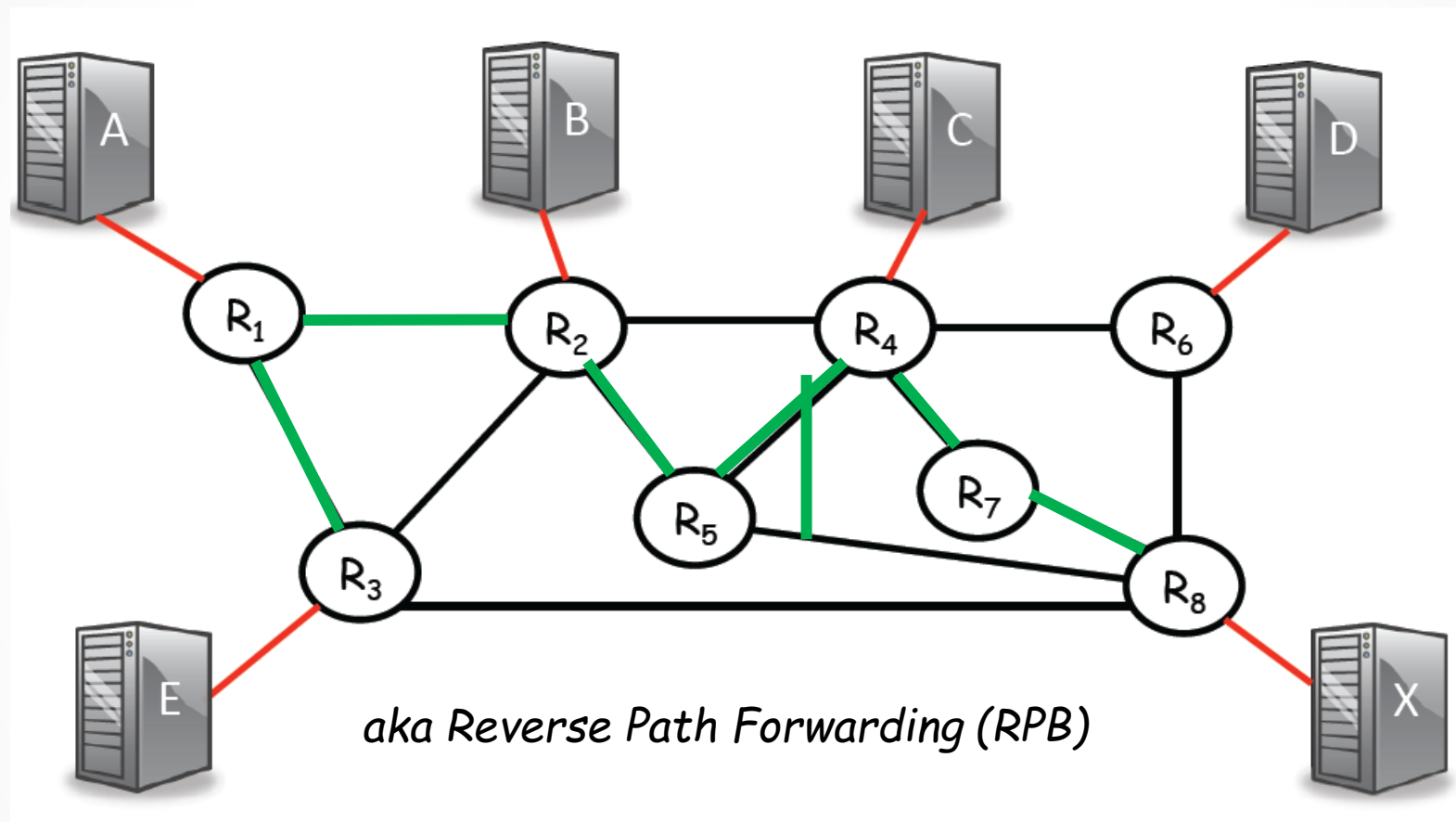
*Задача получила своё название в честь Якоба Штейнера (1796—1863).*



**Дерево наименьшей стоимости для группы**



# Вещание по обратному пути (RPB)



оптимальным деревом для групповой маршрутизации будет дерево с минимальной суммой стоимостей входящих в него линий.

дерево Штейнера



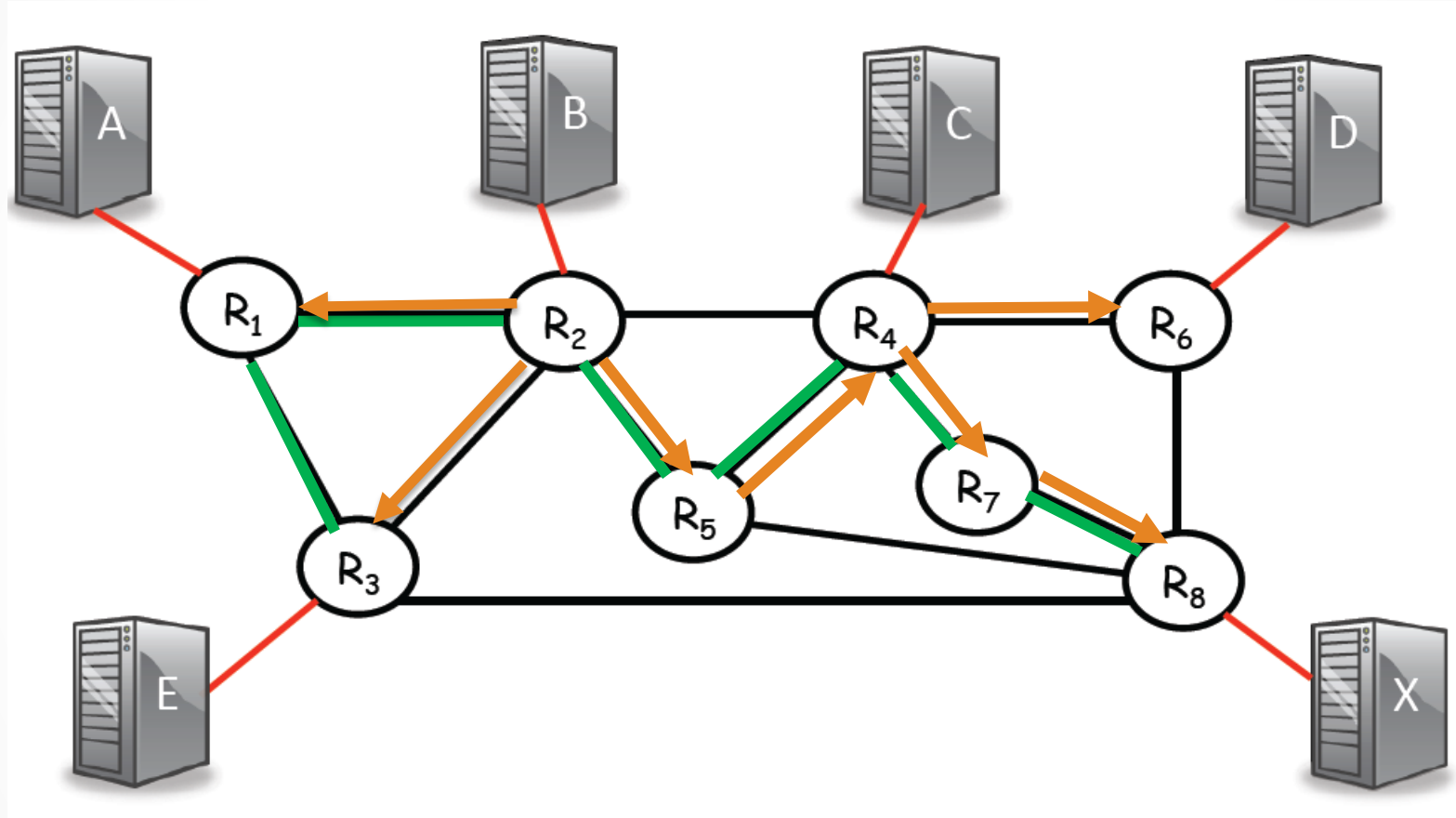
## RPB + обрезка (pruning)

- 1. Каждому хосту пакеты доставляют без циклов*
- 2. Маршрутизаторы через которые нет путей к нужным маршрутизаторам и хостам, шлют сообщения обрезки*
- 3. Результирующее дерево - дерево соединений минимальной стоимости от источника к группе интересующих хостов*





# Одно дерево или несколько ?



Соединяющее дерево для каждого источника свое

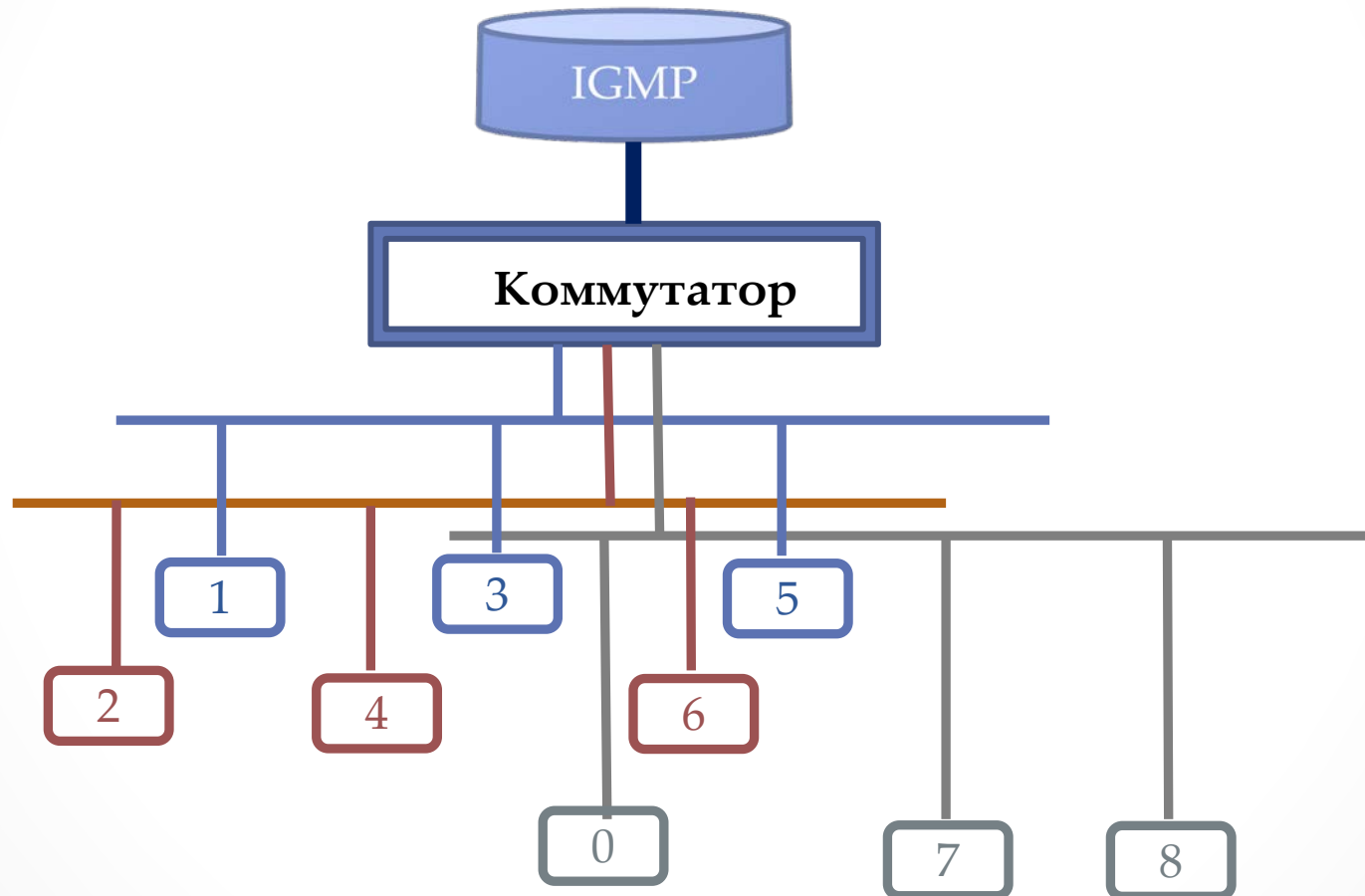


# Адресация и подключение к группе

- *IPv4: сети класса D специально выделены для групповой адресации*
- *IGMP (Internet Group Management Protocol - RFC3376)*
  - *Этот протокол действует между хостом и непосредственно подсоединенным маршрутизатором на уровне L2*
  - *Хосты подписываются/запрашивают получать пакеты определенной группы*
  - *Маршрутизаторы периодически опрашивают хосты к каким группам они хотели бы быть подключенными*
  - *Если отклика нет, то членство в группе прекращается по time\_out (soft-state)*



# IGMP – протокол управления группой





# Групповая маршрутизация на практике

- **Актуальность групповой маршрутизации на практике постоянно возрастает**
  - по большей части коммуникации индивидуализированы
  - ранние реализации были не эффективны
  - сегодня в основном используются для IP TV и быстрой рассылки
  - используется отдельными приложениями

## **Интересные вопросы:**

*Как сделать групповое взаимодействие надежным?*

*Как реализовать управление потоком?*

*Как поддерживать разную скорость работы с разными клиентами?*

*Как обеспечить конфиденциальность при групповом взаимодействии?*

*Как быть когда коммутация на L2 и маршрутизация на L3 независимы?*



# Маршрутизация на L2

Введение в компьютерные сети

чл.-корр. РАН Смелянский Р.А.

Кафедра АСВК

ф-т ВМК МГУ



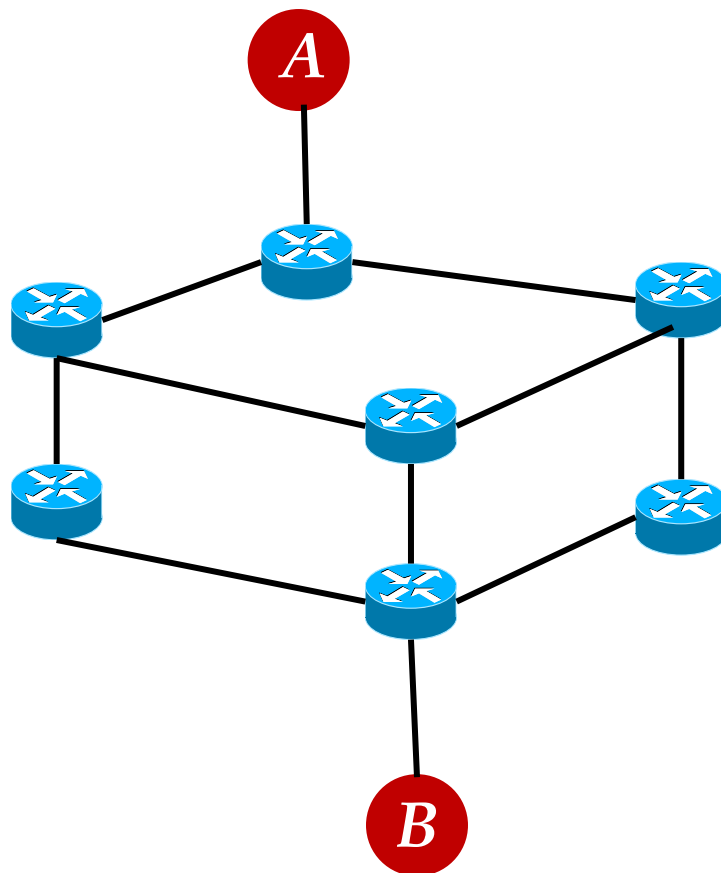


# Ethernet коммутатор (Learning Switch)

1. *Анализирует заголовки каждого поступающего кадра*
2. *Если DA есть в таблице коммутации, передать кадр на надлежащий порт-выход*
3. *Если DA нет в таблице коммутации, разослать кадр по всем портам за исключением того, по которому он пришел*
4. *Коммутатор «изучает» сеть - Таблица коммутации пополняется за счет изучения адресов SA поступающих пакетов*

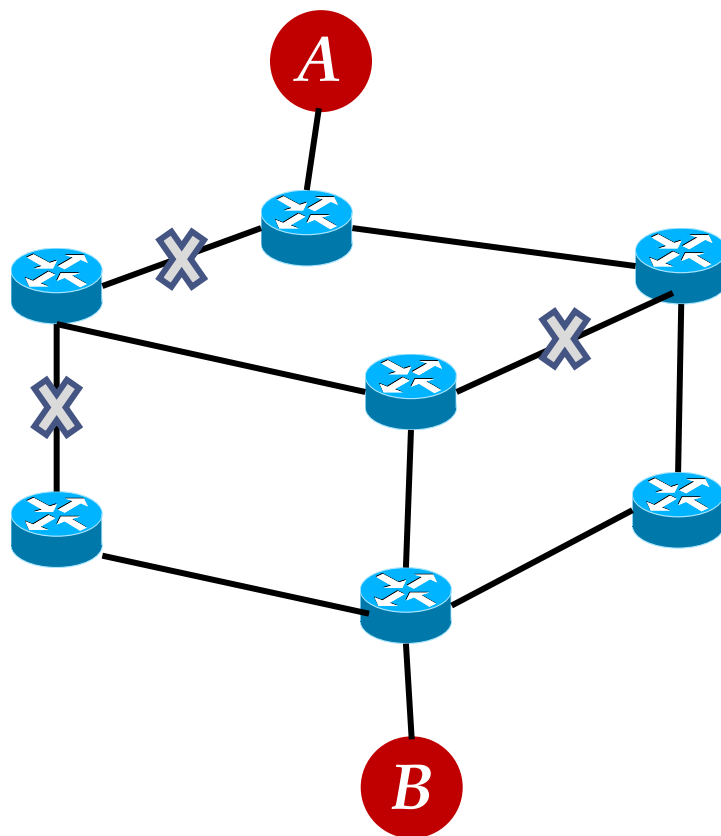


# «Изучение» может зацикливаться





# «Изучение» может зацикливаться





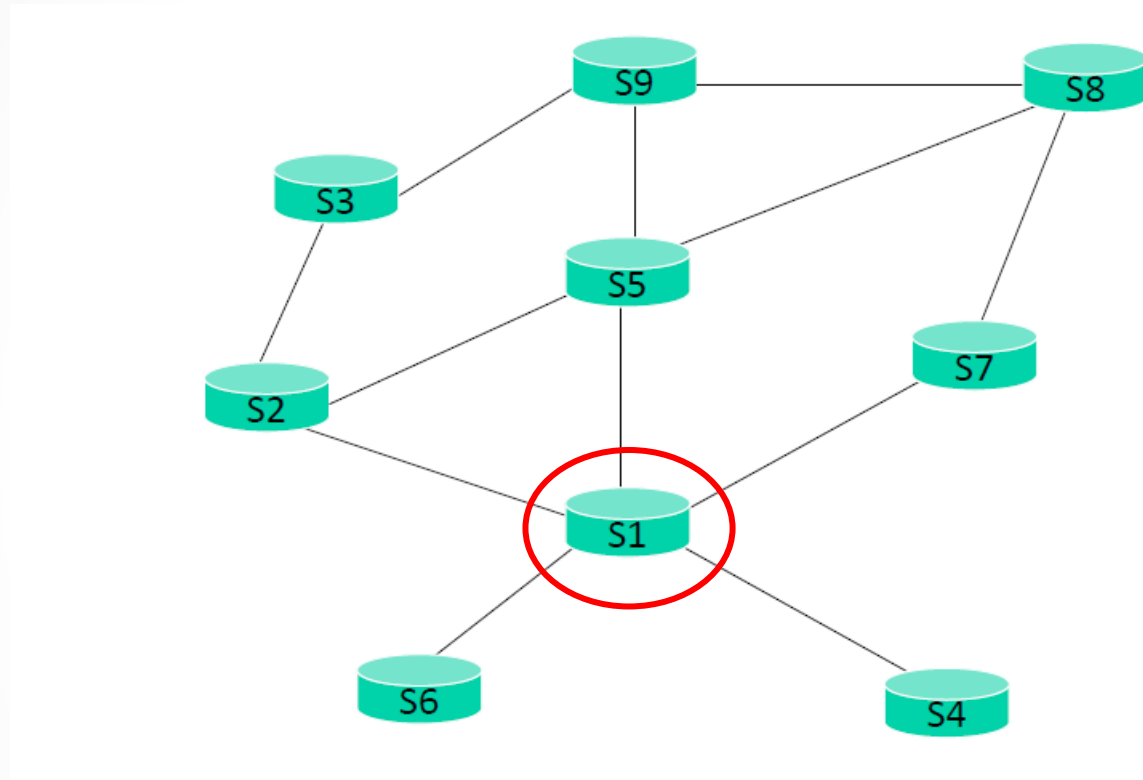


# Предотвращение зацикливаний (протокол соединяющего дерева - spanning tree protocol)

- *Топология коммутаторов - граф*
- *Протокол STP находит подграф (остовное дерево), в котором все вершины соединены без циклов*
  - *соединяющее - к любому коммутатору есть путь*
  - *дерево - нет циклов*
- *STP распределенный протокол*
  - *какой из коммутаторов - корень дерева*
  - *каким портам разрешено рассылать кадры вдоль дерева ?*



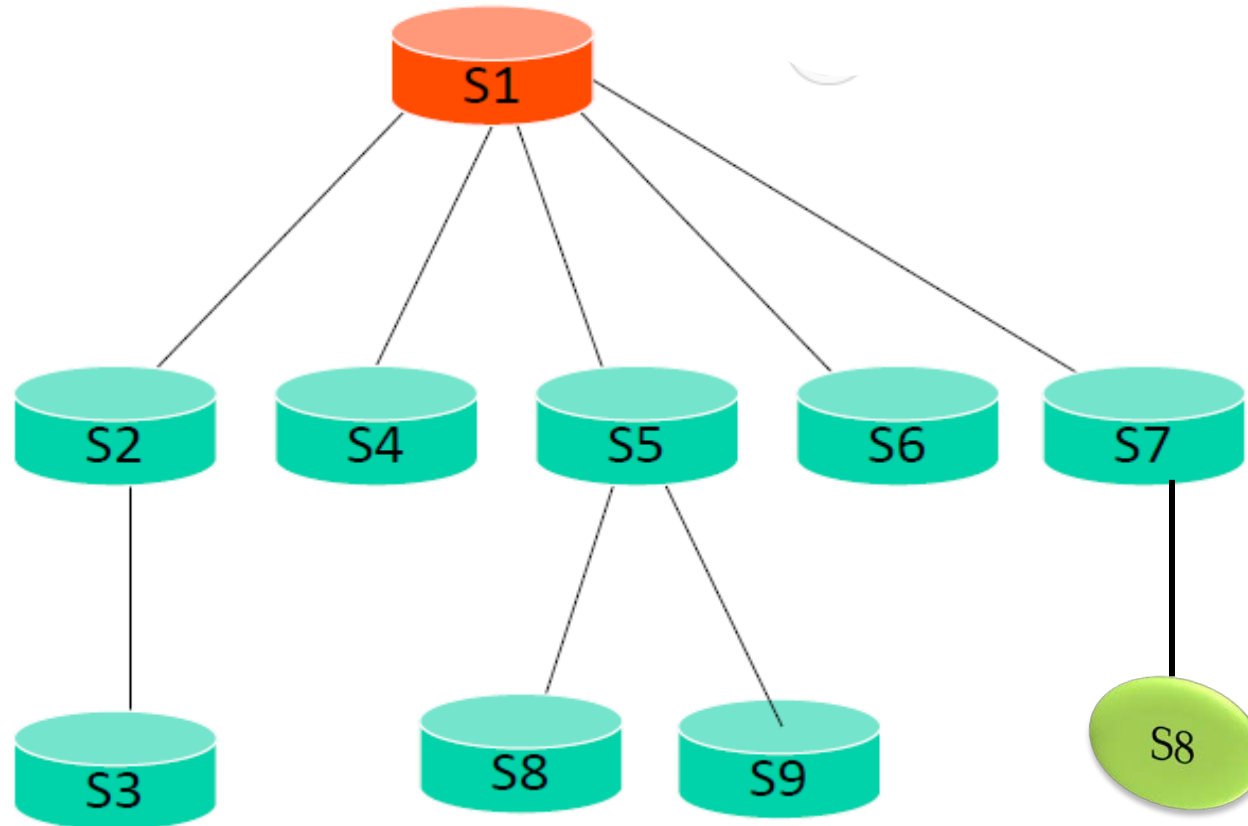
# Пример соединяющего дерева



- *выбираем корень*
- *кадр коммутируется на тот порт, который ведет к корню за наименьшим число скачков (hop)*



# Соединяющее дерево для нашего примера



# Как это работает



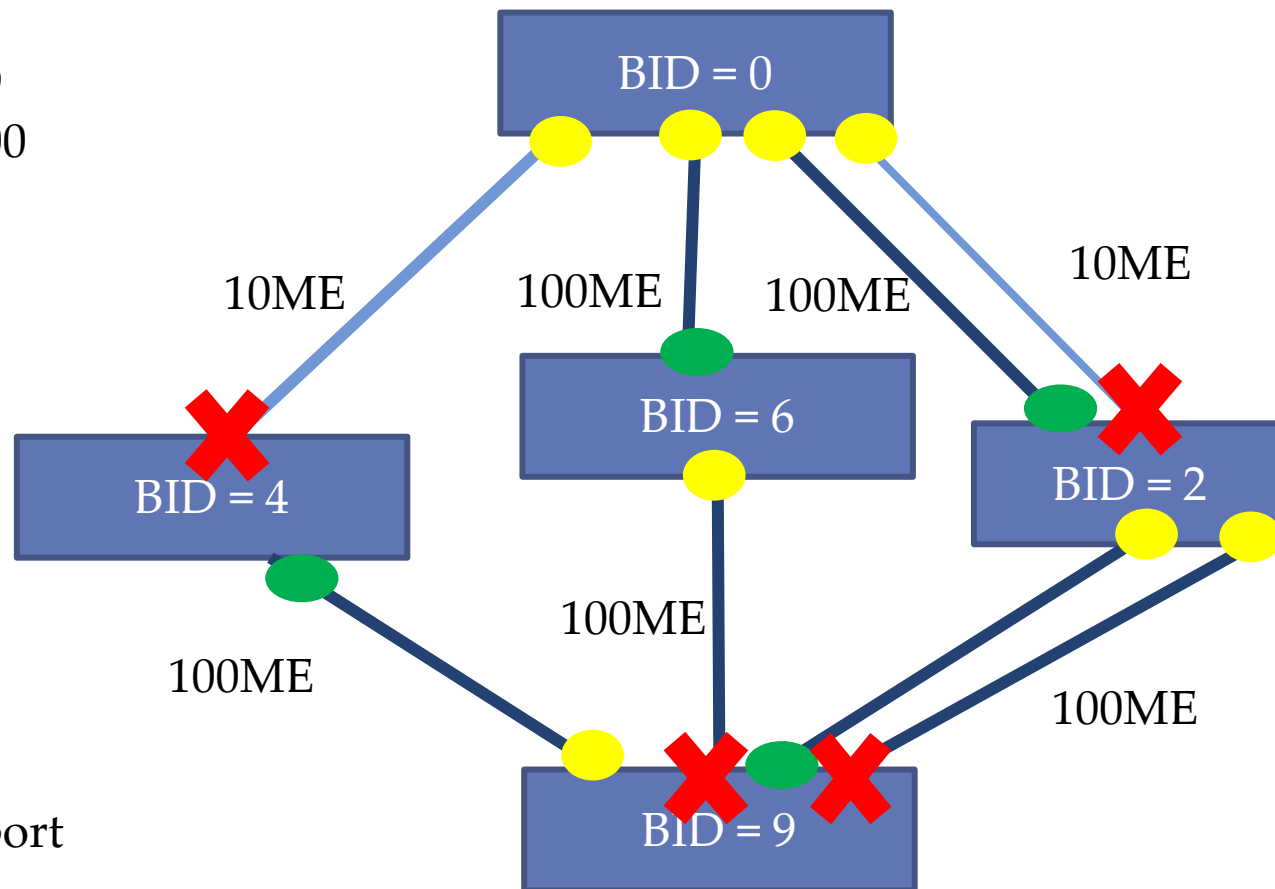
После включения коммутаторов в сеть, по умолчанию каждый коммутатор считает себя корневым (*root*).

- Каждый коммутатор начинает посылать по всем портам конфигурационные Hello [BPDU](#) пакеты раз в 2 секунды. (BPDU (Bridge PDU) – ID отправителя, ID корня, расстояние от отправителя до корня)). Изначально все считают себя корнем (расстояние = 0).
- Если коммутатор получает [BPDU](#) с идентификатором Bridge ID меньшим, чем свой собственный, он прекращает генерировать свои BPDU и начинает ретранслировать BPDU с этим идентификатором. Таким образом в конце концов в этой сети Ethernet остаётся только один коммутатор, который продолжает генерировать и передавать собственные BPDU. Он и становится корневым (*root bridge*).
- Остальные коммутаторы ретранслируют BPDU корневого, добавляя в них собственный идентификатор и увеличивая счётчик пути (*path cost*).
- Для каждого сегмента сети, к которому присоединены два и более портов коммутаторов, происходит определение *rootport*, потом *designated port* – порта, через который BPDU, приходящие от корневого коммутатора, попадают в этот сегмент.
- После этого все порты в сегментах, к которым присоединены 2 и более портов коммутаторов, блокируются за исключением *root port* и *designated port*.
- Корневой хост продолжает посылать свои Hello BPDU раз в 2 секунды.



# Построение ST

100ME = 19  
 10 ME = 100  
 1 GE = 4  
 10GE = 2



 - Root port

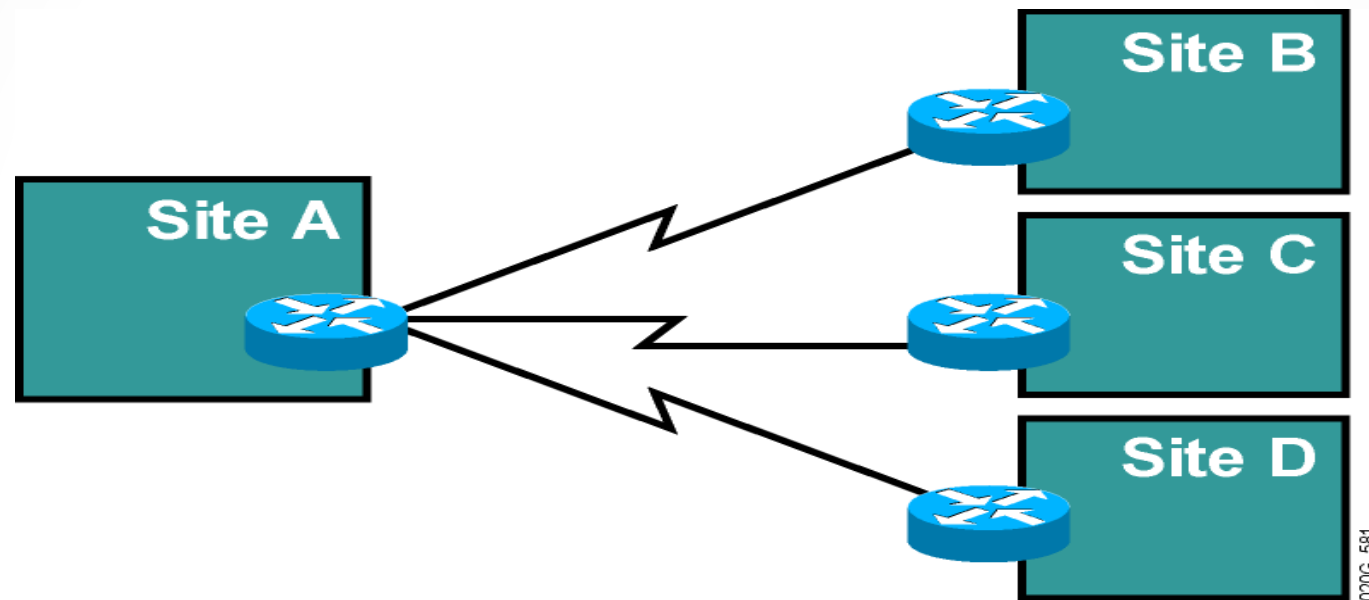
 - Designated port

## Bridge Protocol Data Unit (BPDU)

Название поля	Размер поля
Protocol Identifier	2 байта
Protocol Version Identifier	1 байт
BPDU Type	1 байт
Flags	1 байт
Root Identifier	8 байт
Root Path Cost	4 байта
Bridge Identifier	8 байт
Port Identifier	2 байта
Message Age	2 байта
Max Age	2 байта
Hello Time	2 байта
Forward Delay	2 байта



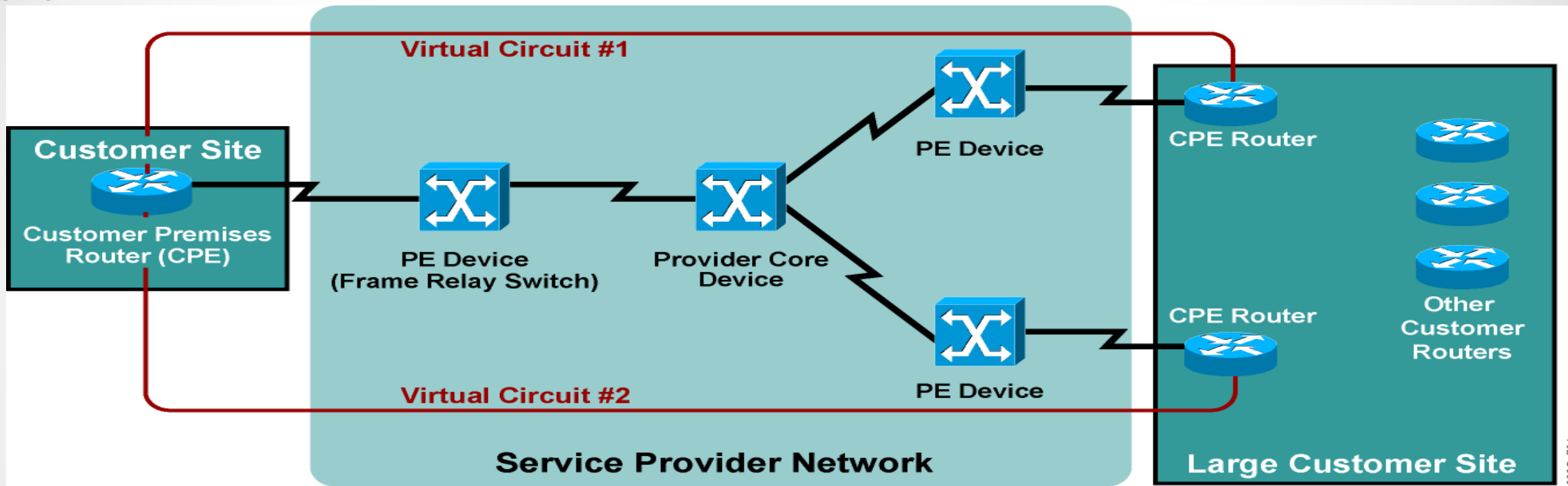
# MPLS: Традиционные сети



- В сетях на основе традиционных маршрутизаторов сайты клиентов подключены через маршрутизаторы через выделенные каналы «точка-точка».



# MPLS: Virtual Private Networks



0206\_784

- VPN сети заменяют выделенные каналы «точка-точка» виртуальными (эмулированными) каналами «точка-точка», использующими общую инфраструктуру сетевого оператора.



# Заключение

- *Есть много способов маршрутизации пакетов в сети*
- *Для расчета маршрута используют протокол маршрутизации*
- *Алгоритмы маршрутизации часто используют соединяющие деревья с минимальной стоимостью до места назначения*
- *Интернет имеет иерархическую структуру из AS*
- *Внутри AS действует один из IGP (OSPF, RIP, IS-IS) протоколов*
- *Маршрутизация между AS по протоколу BGP-4 (EBGP)*
- *Если у AS несколько BGP шлюзов, то эти шлюзы связаны по протоколу IBGP*
- *AS состоит из областей. Внутри каждой области маршрутизация по OSPF, между областями через смежные (designated) маршрутизаторы*
- *Маршрутизация с множественными путями позволяет распределять нагрузку по нескольким линиям одновременно*
- *Групповая маршрутизация обеспечивает доставку сразу нескольким хостам (здесь проблема построения дерева Штейнера)*