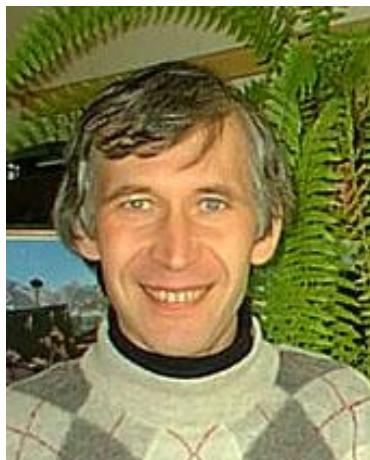


МГУ ВМК



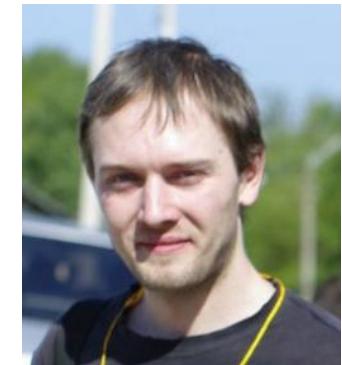
МАТКИБ

# Дискретные модели и задачи управления компьютерными сетями



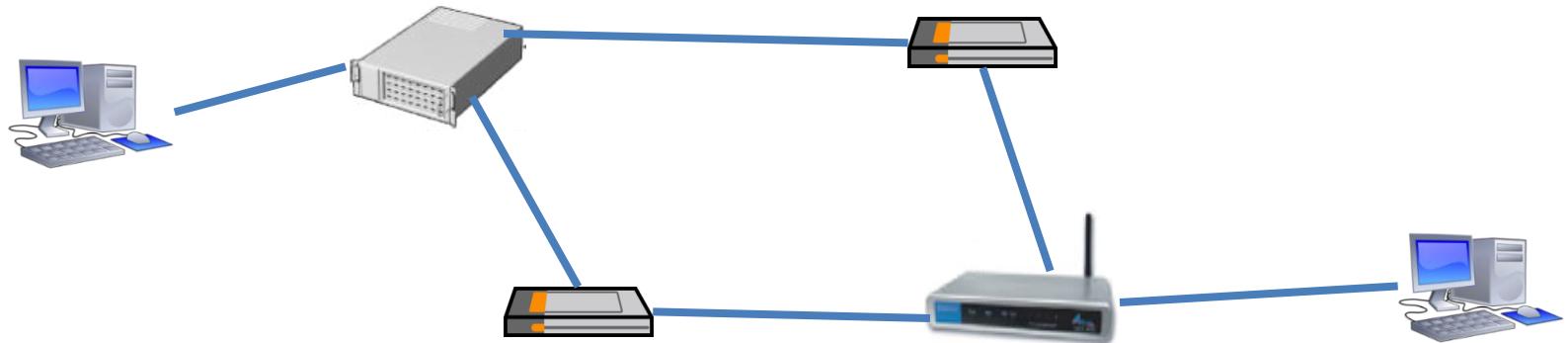
Захаров В.А.

Подымов В.В.



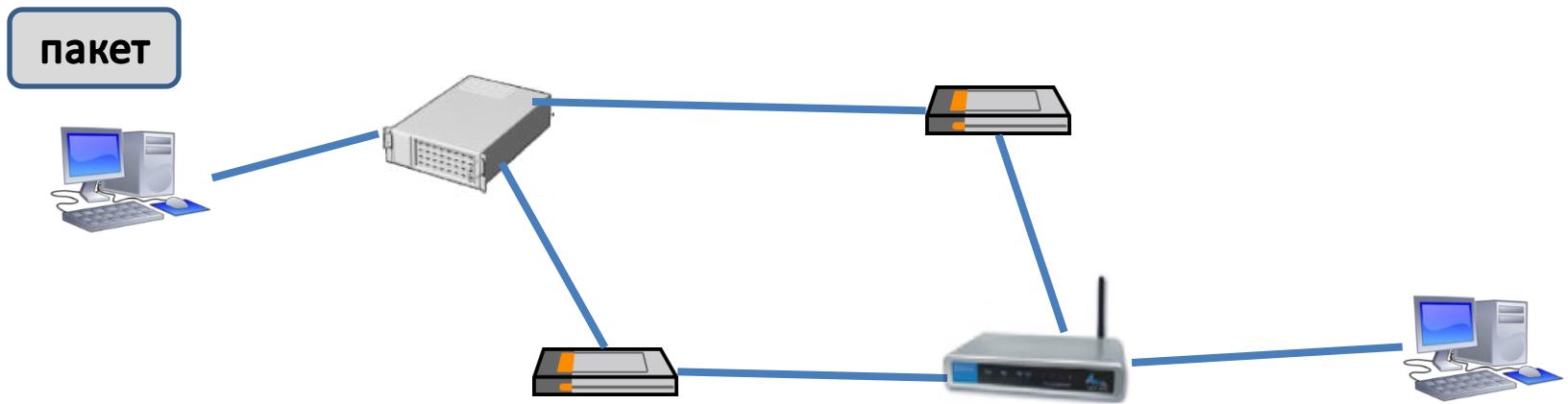
# Классические сети

---

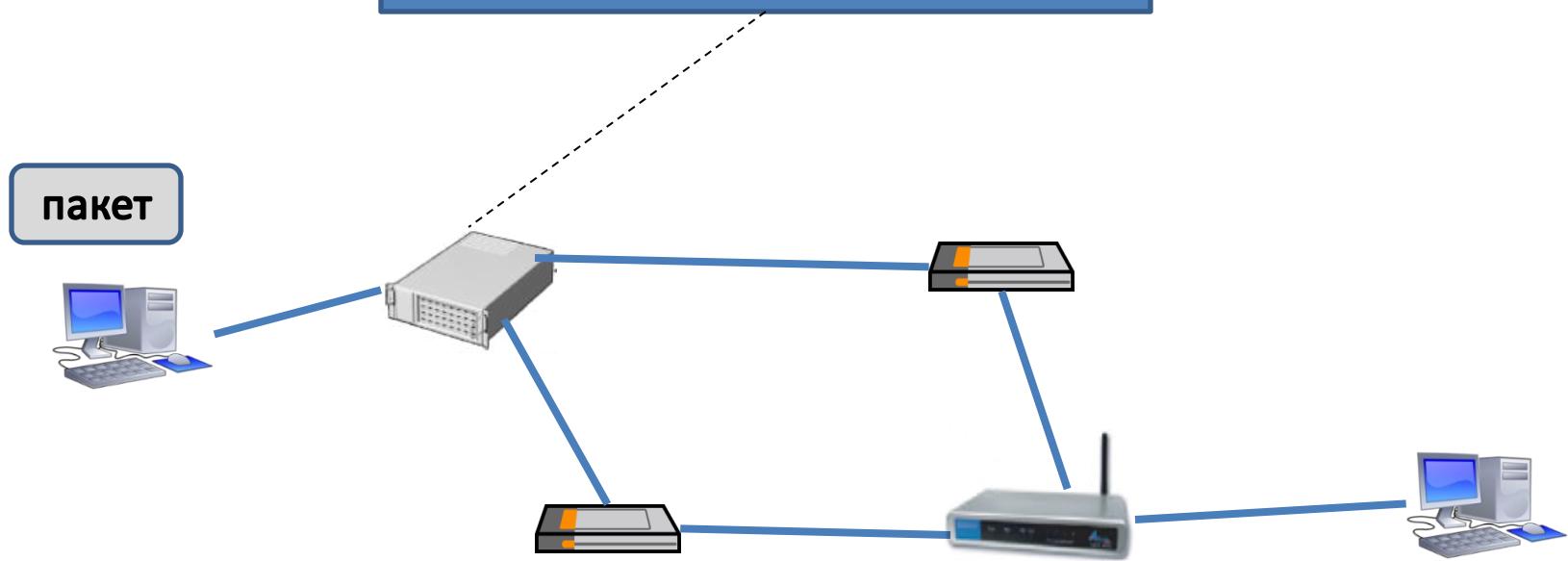
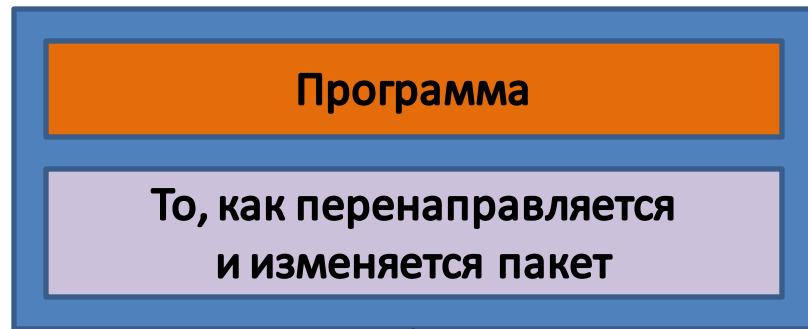


# Классические сети

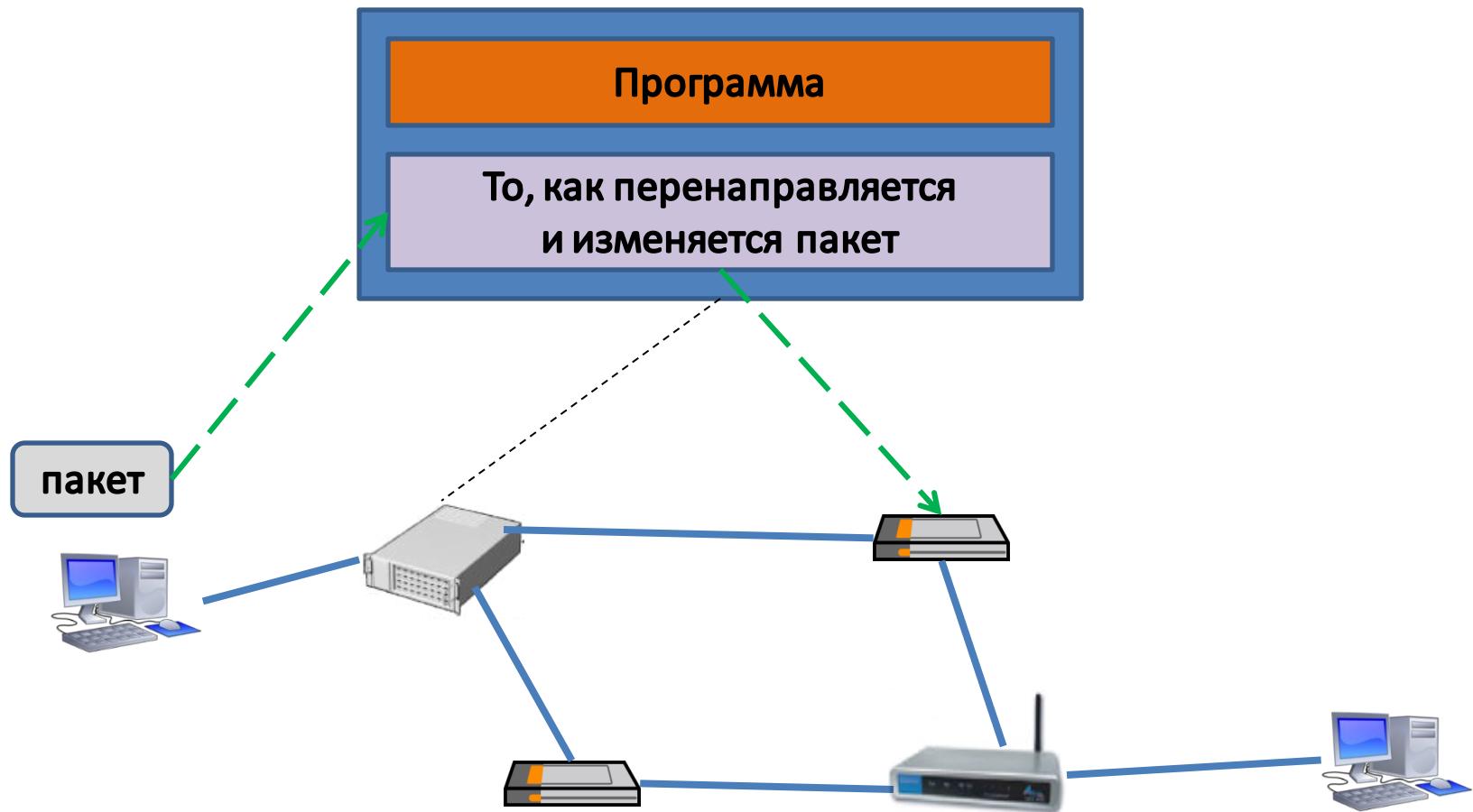
---



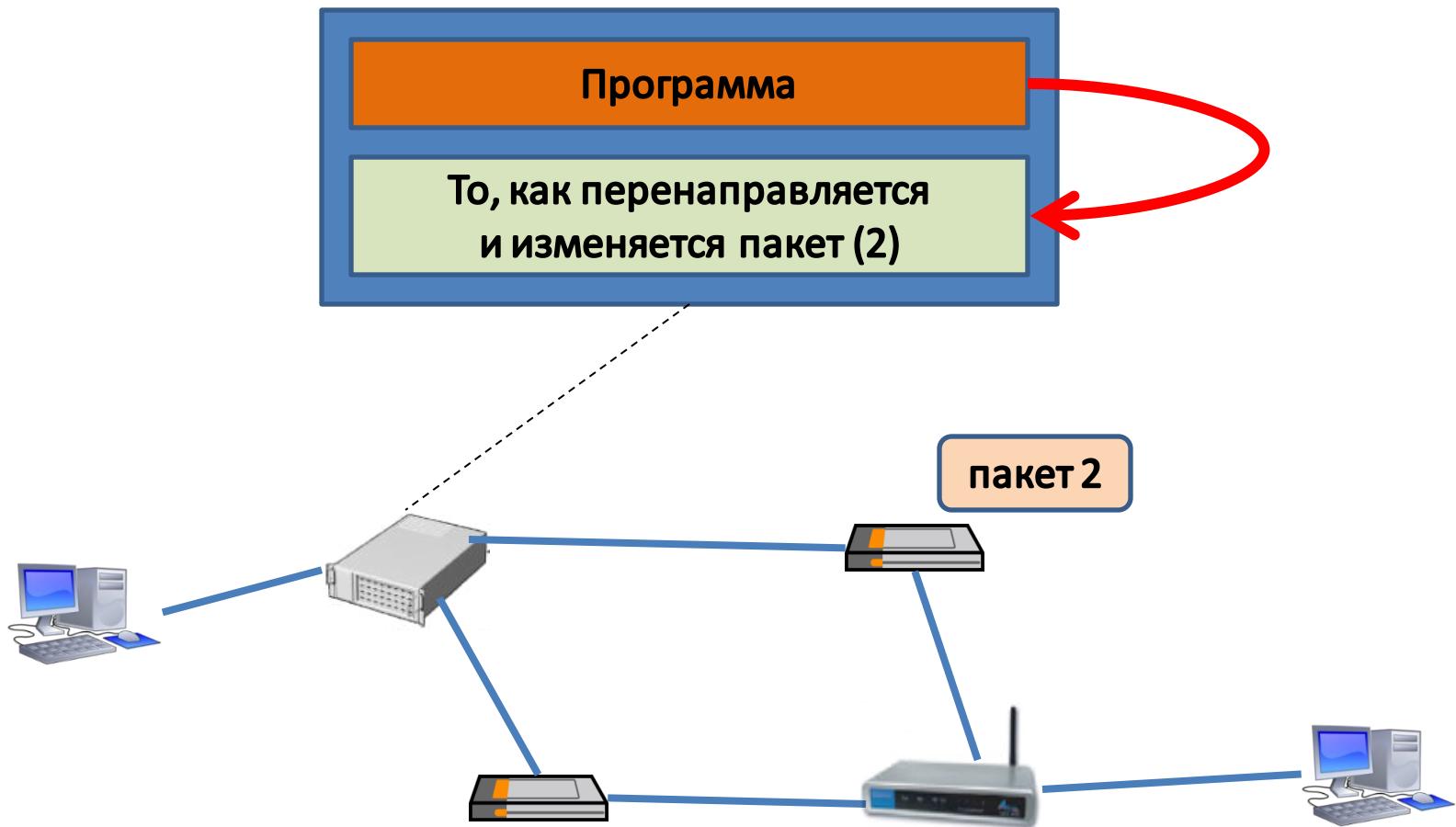
# Классические сети



# Классические сети

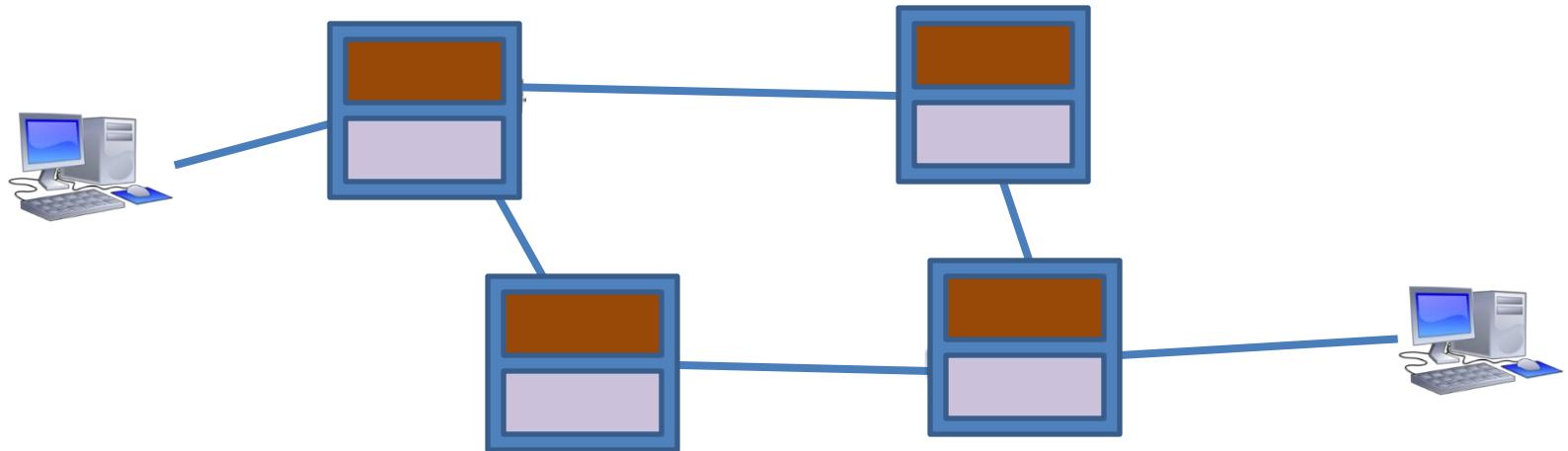


# Классические сети



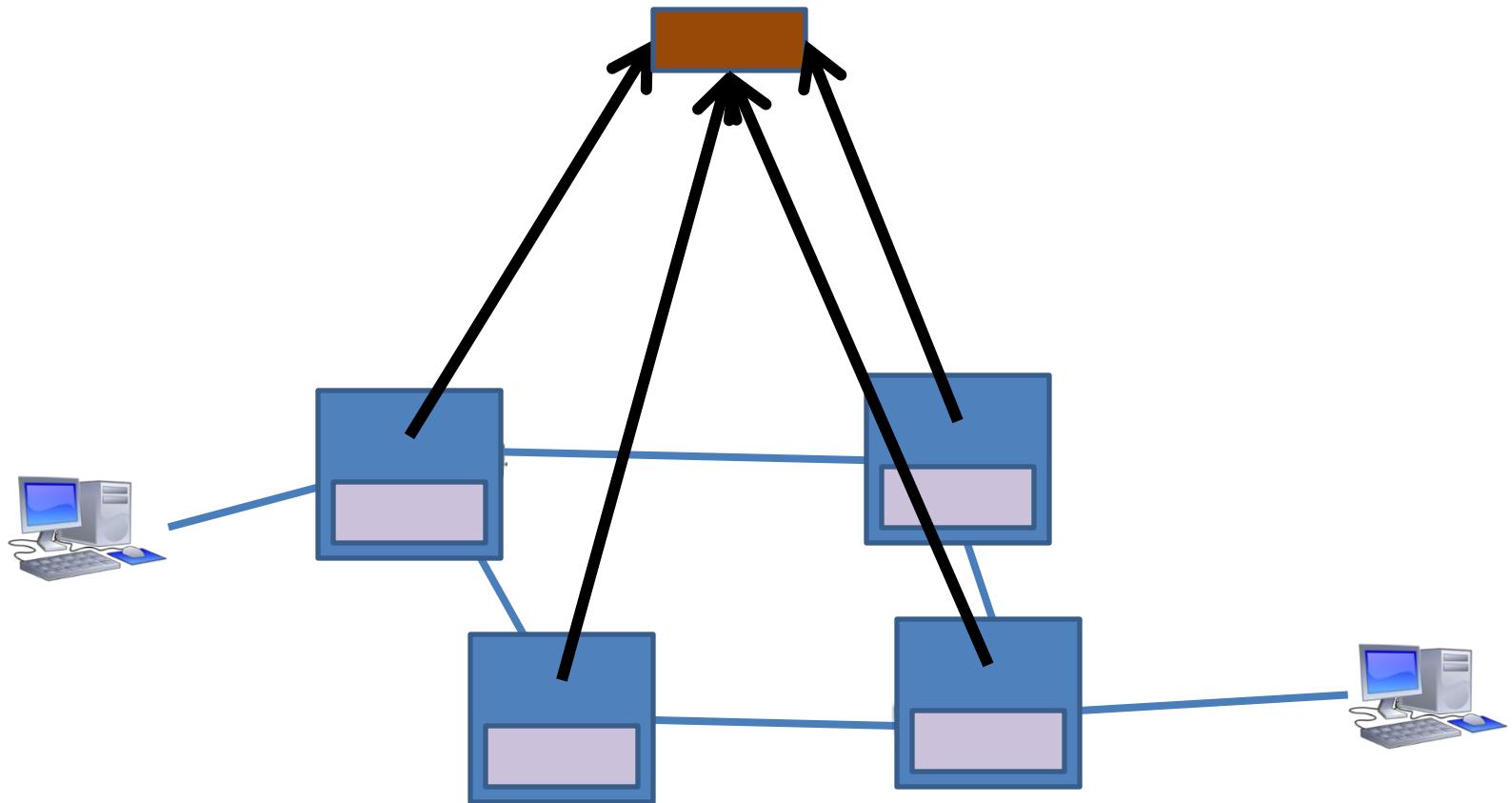
# Классические сети

---



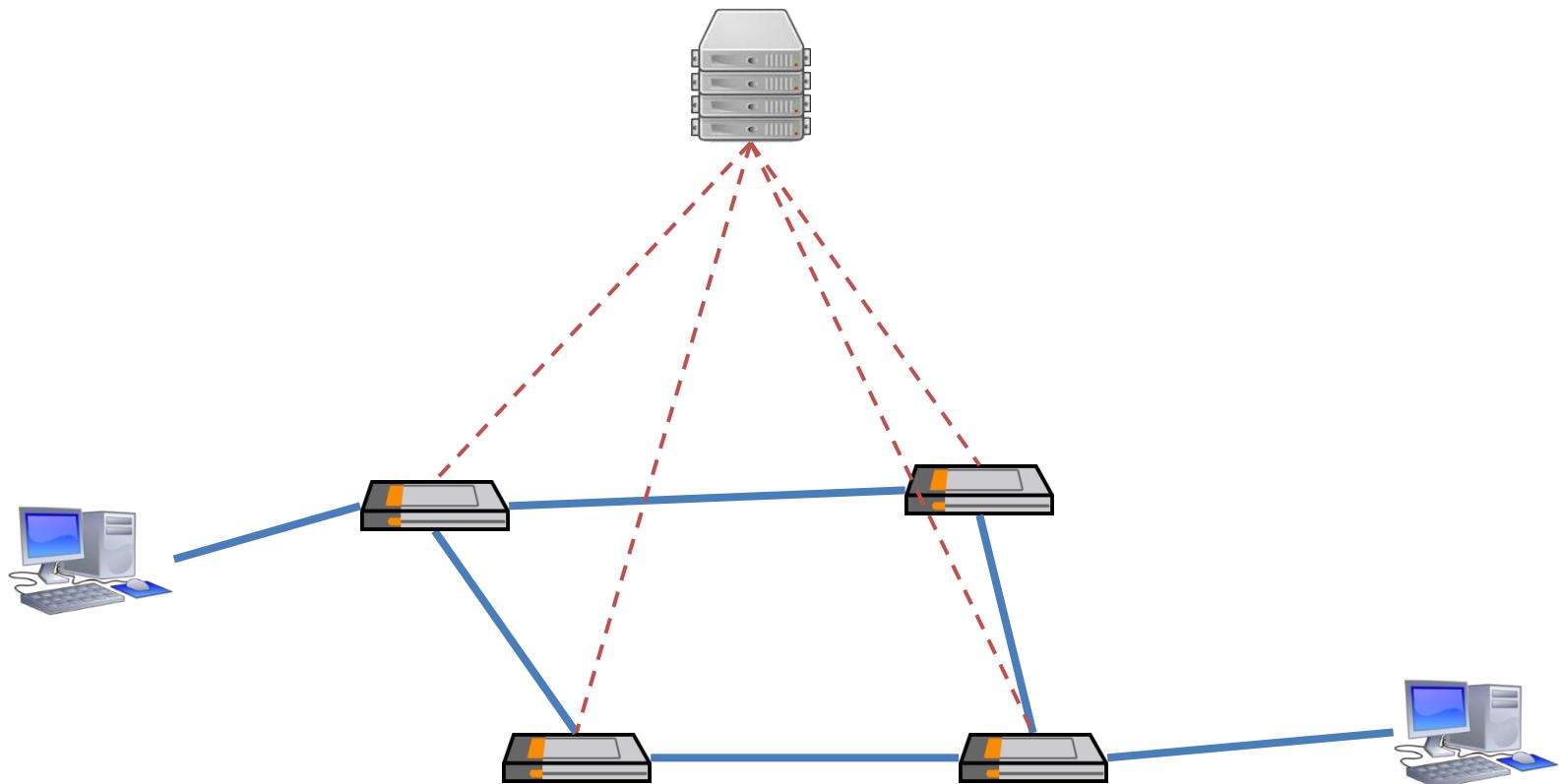
# Классические сети

---

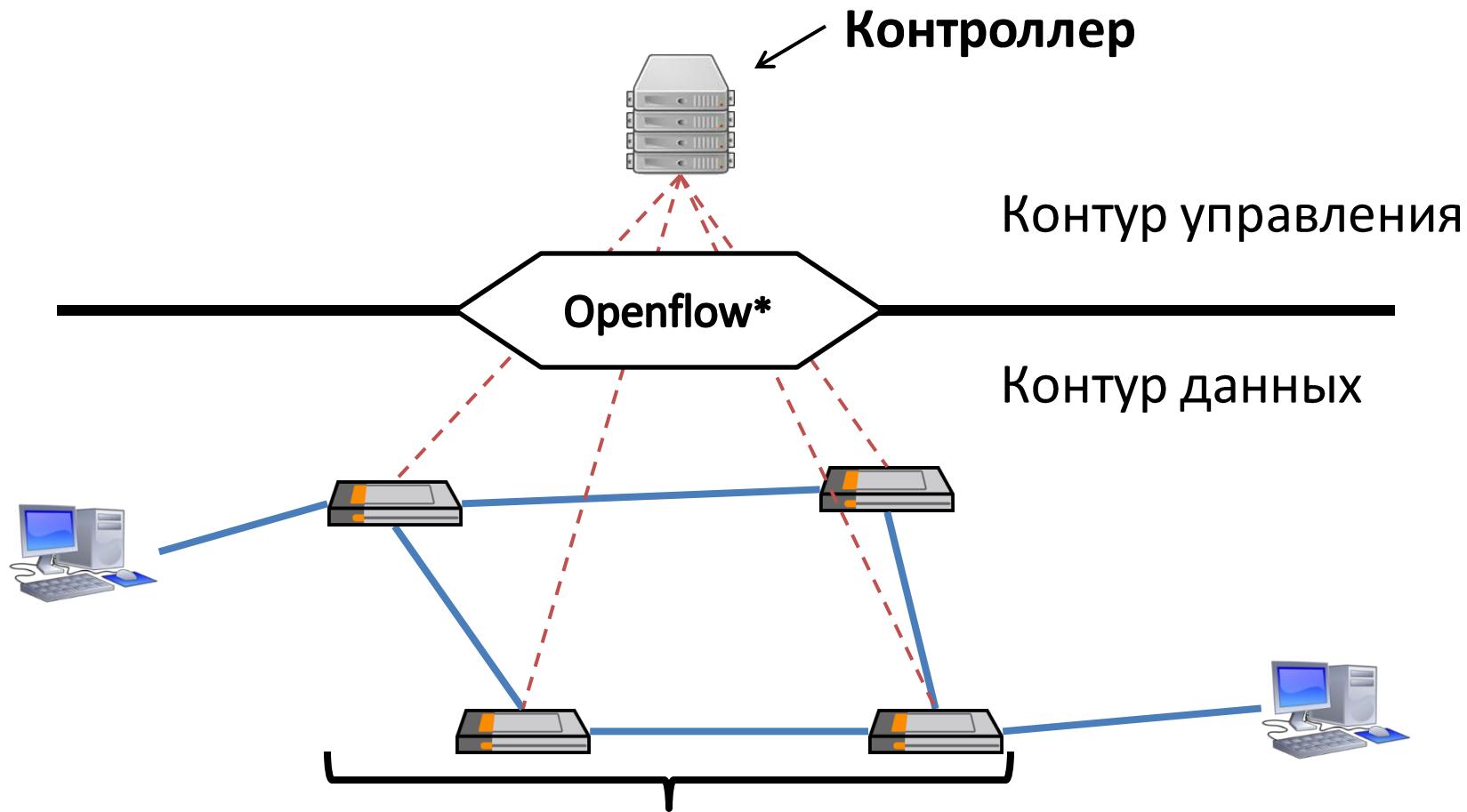


# Программно-конфигурируемые сети (ПКС)

---

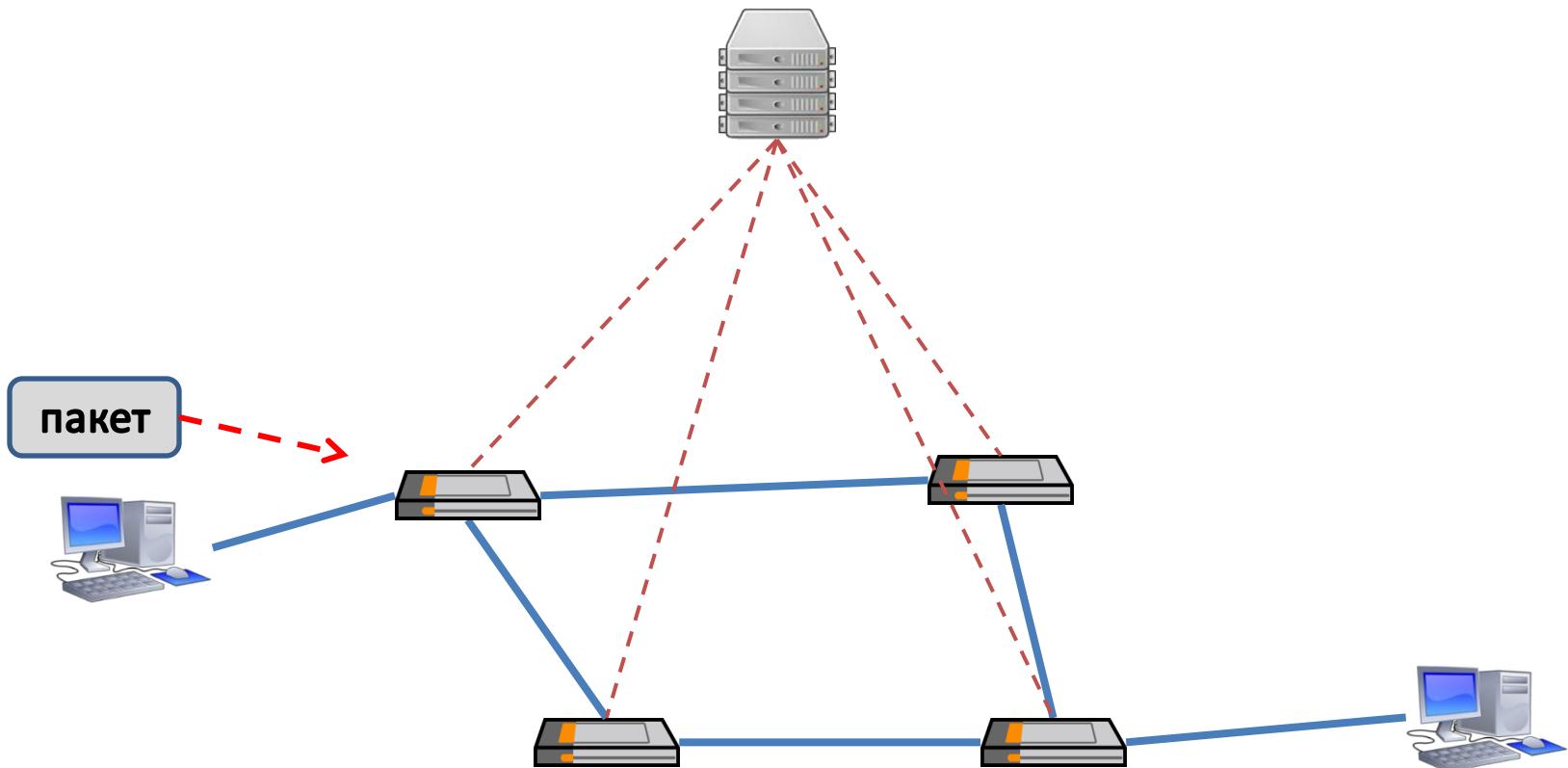


# Программно-конфигурируемые сети (ПКС)

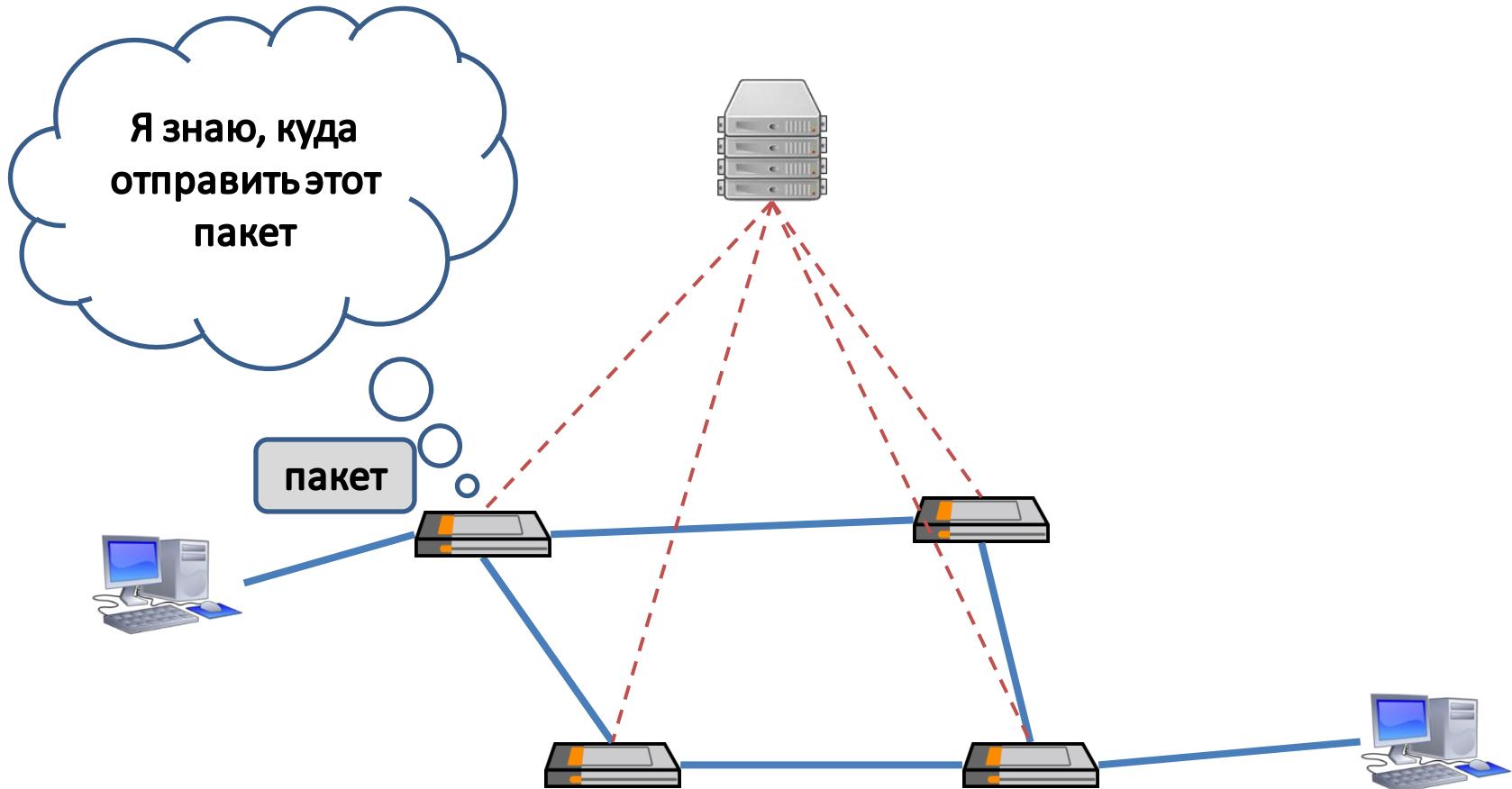


\* Это самый популярный протокол, используемый в ПКС

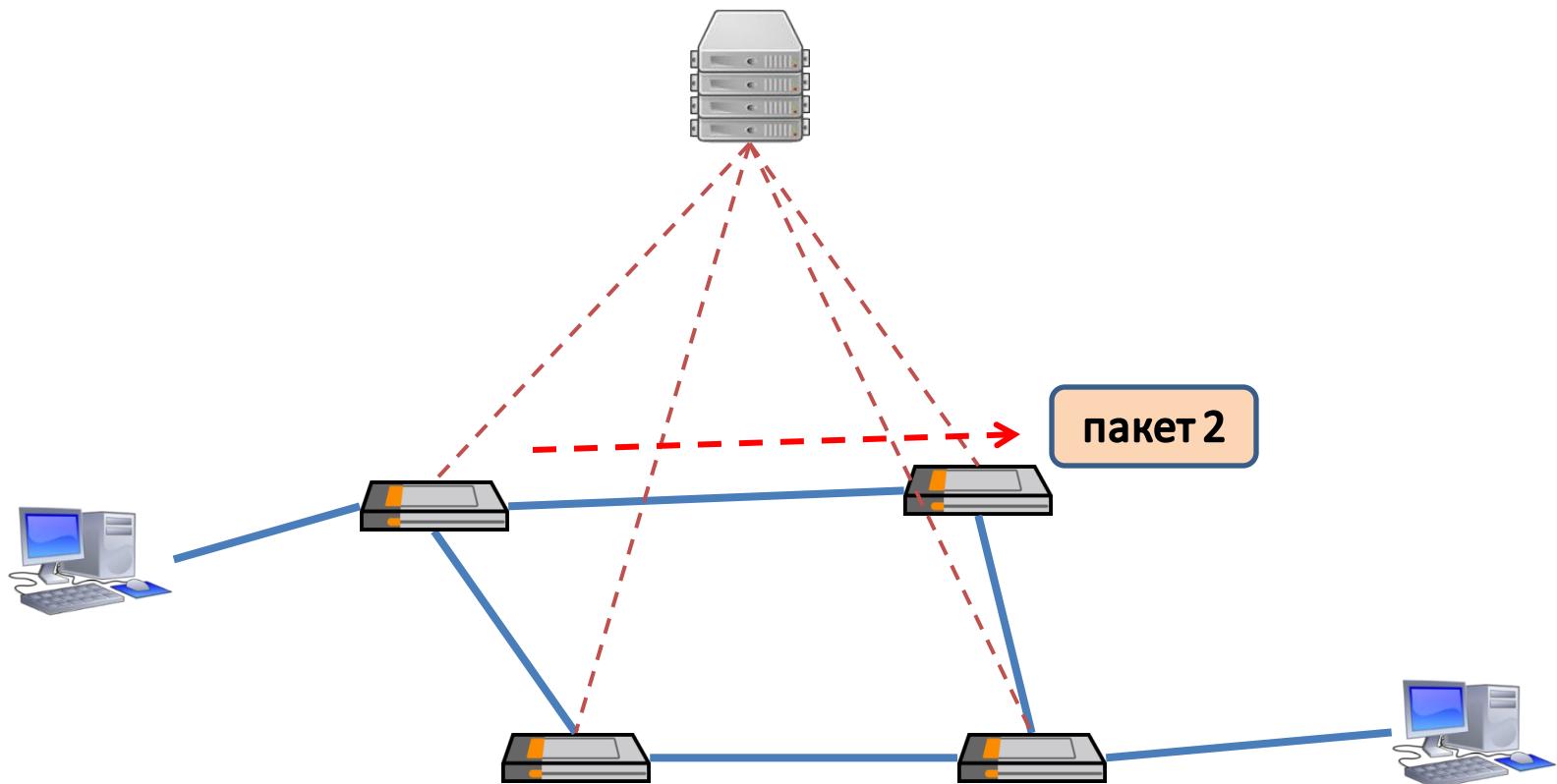
# Программно-конфигурируемые сети (ПКС)



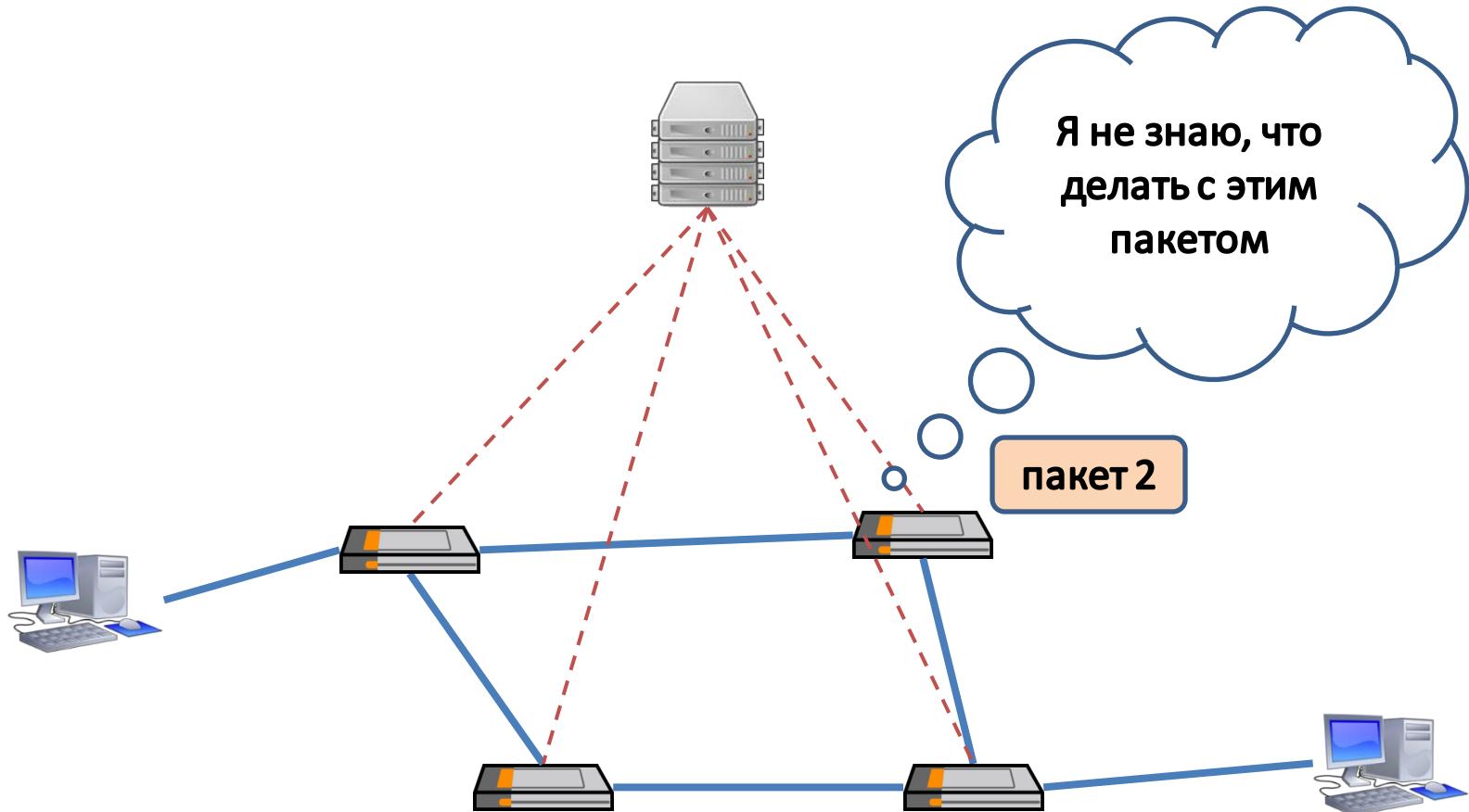
# Программно-конфигурируемые сети (ПКС)



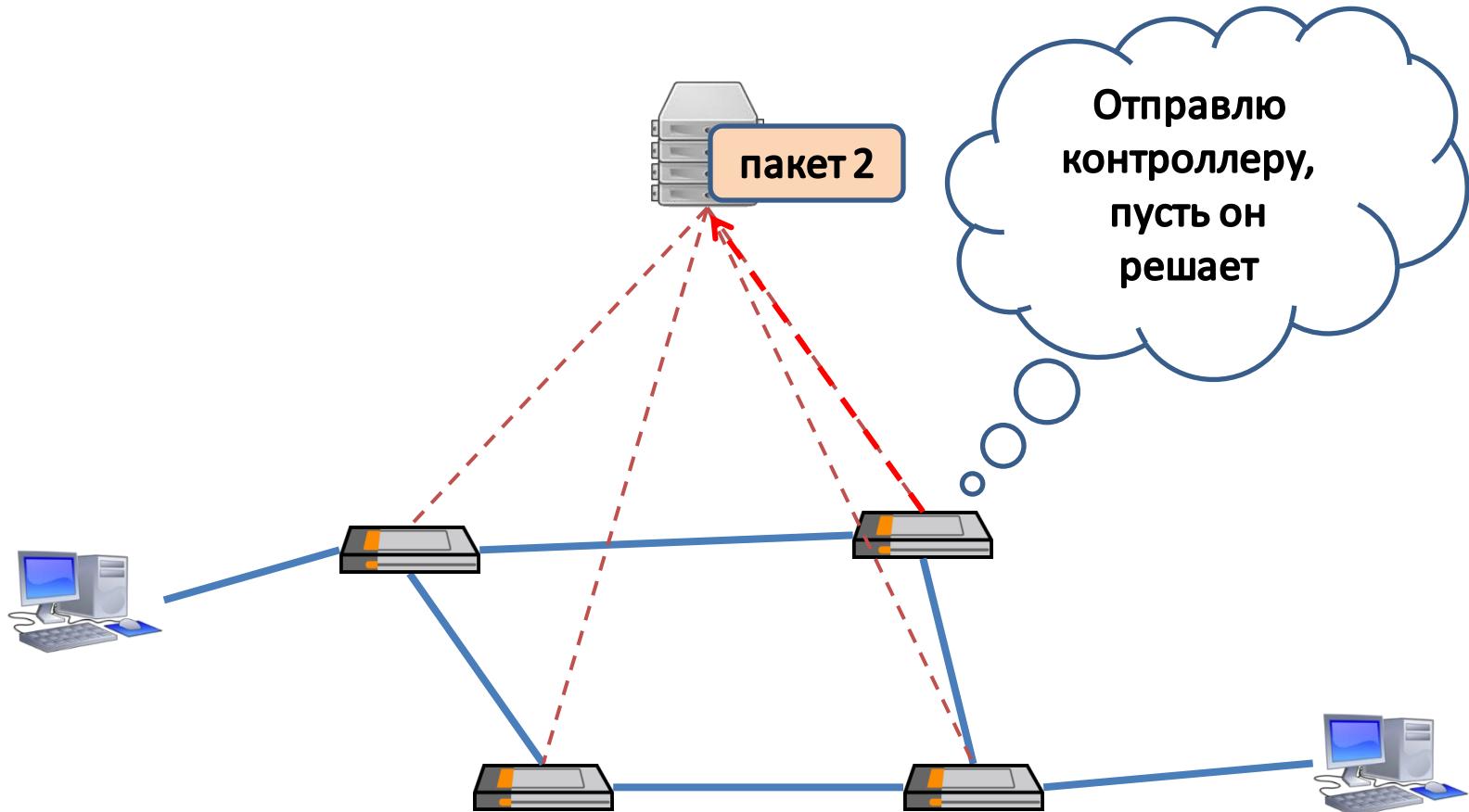
# Программно-конфигурируемые сети (ПКС)



# Программно-конфигурируемые сети (ПКС)

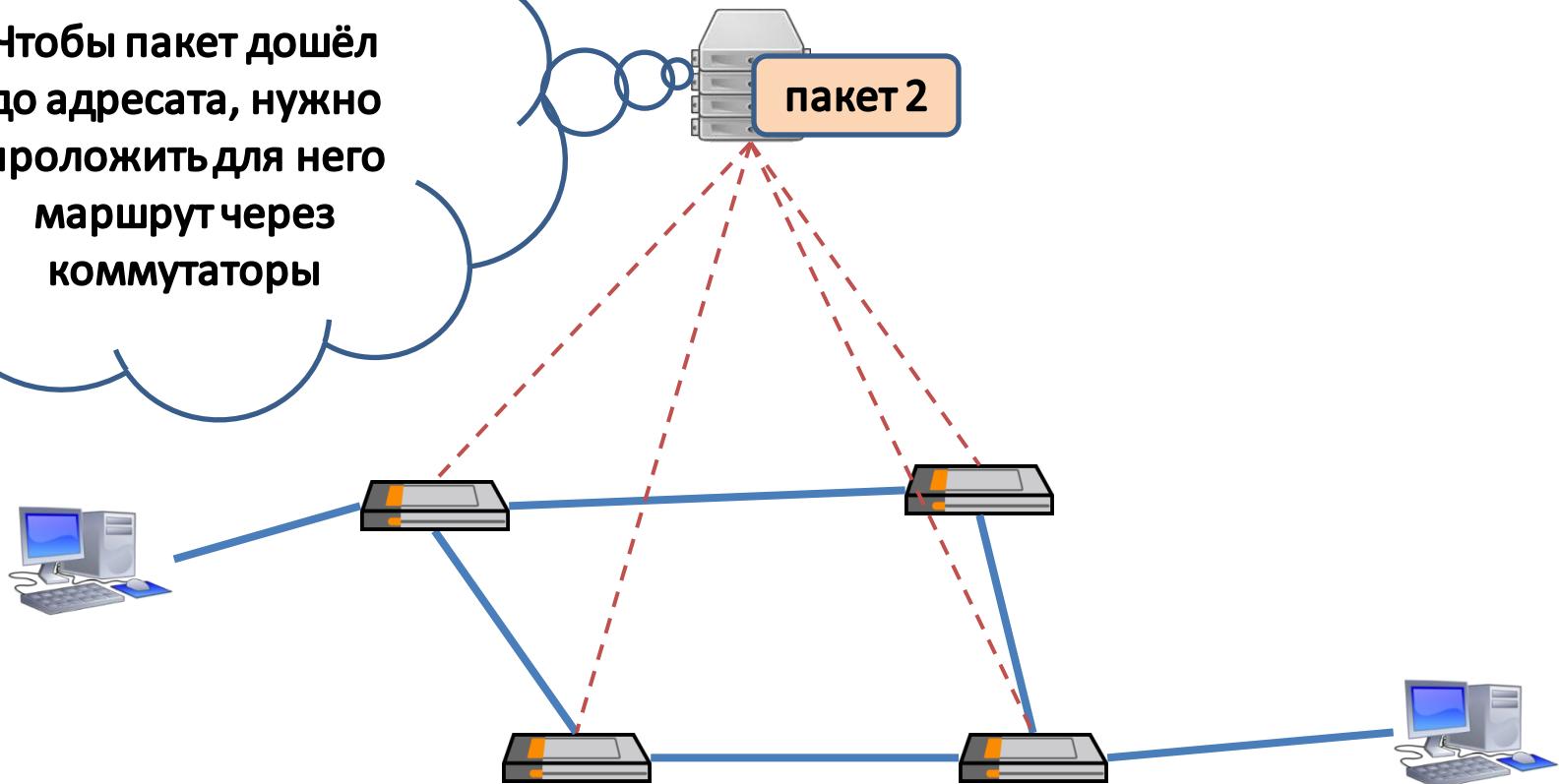


# Программно-конфигурируемые сети (ПКС)

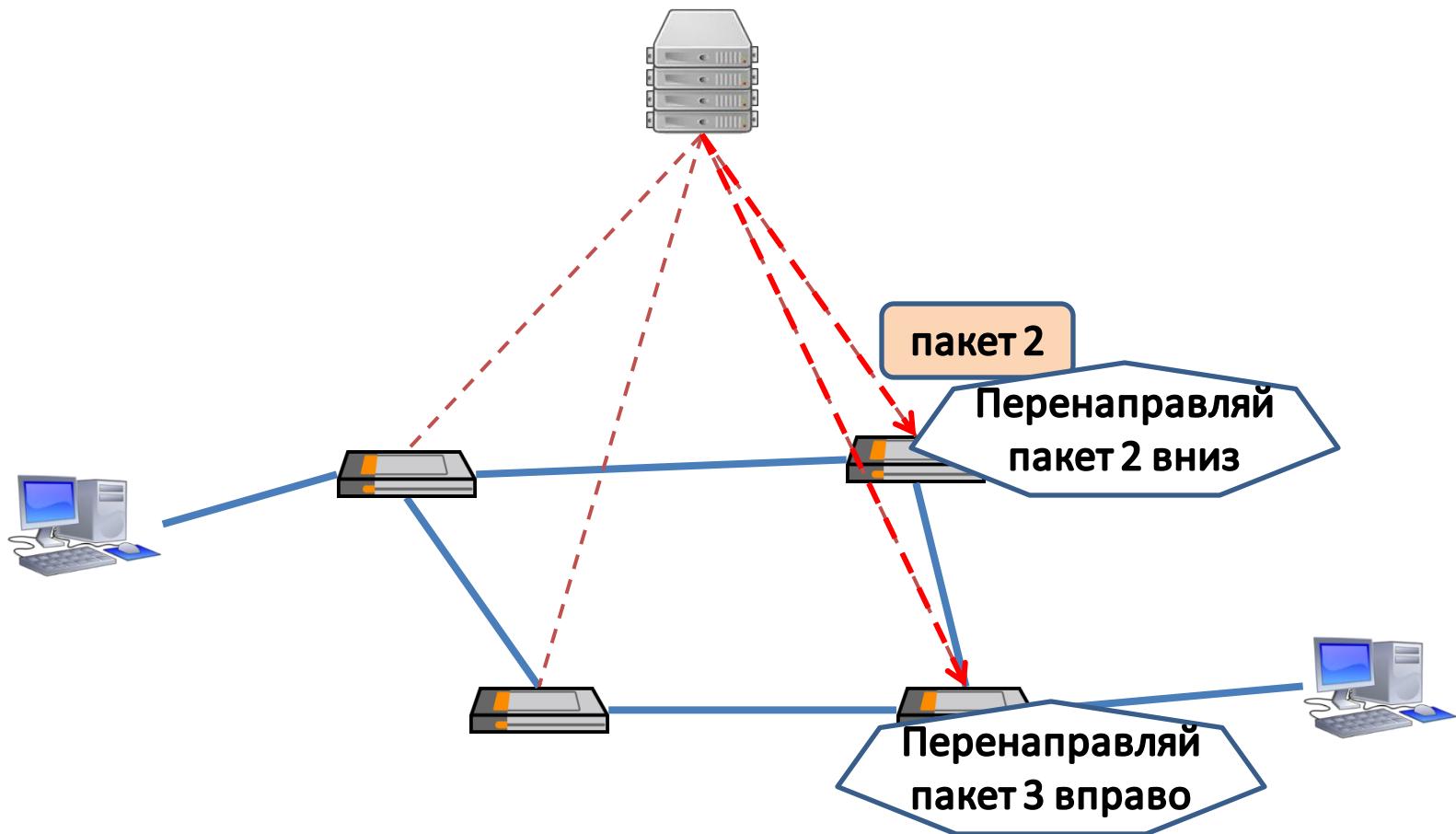


# Программно-конфигурируемые сети (ПКС)

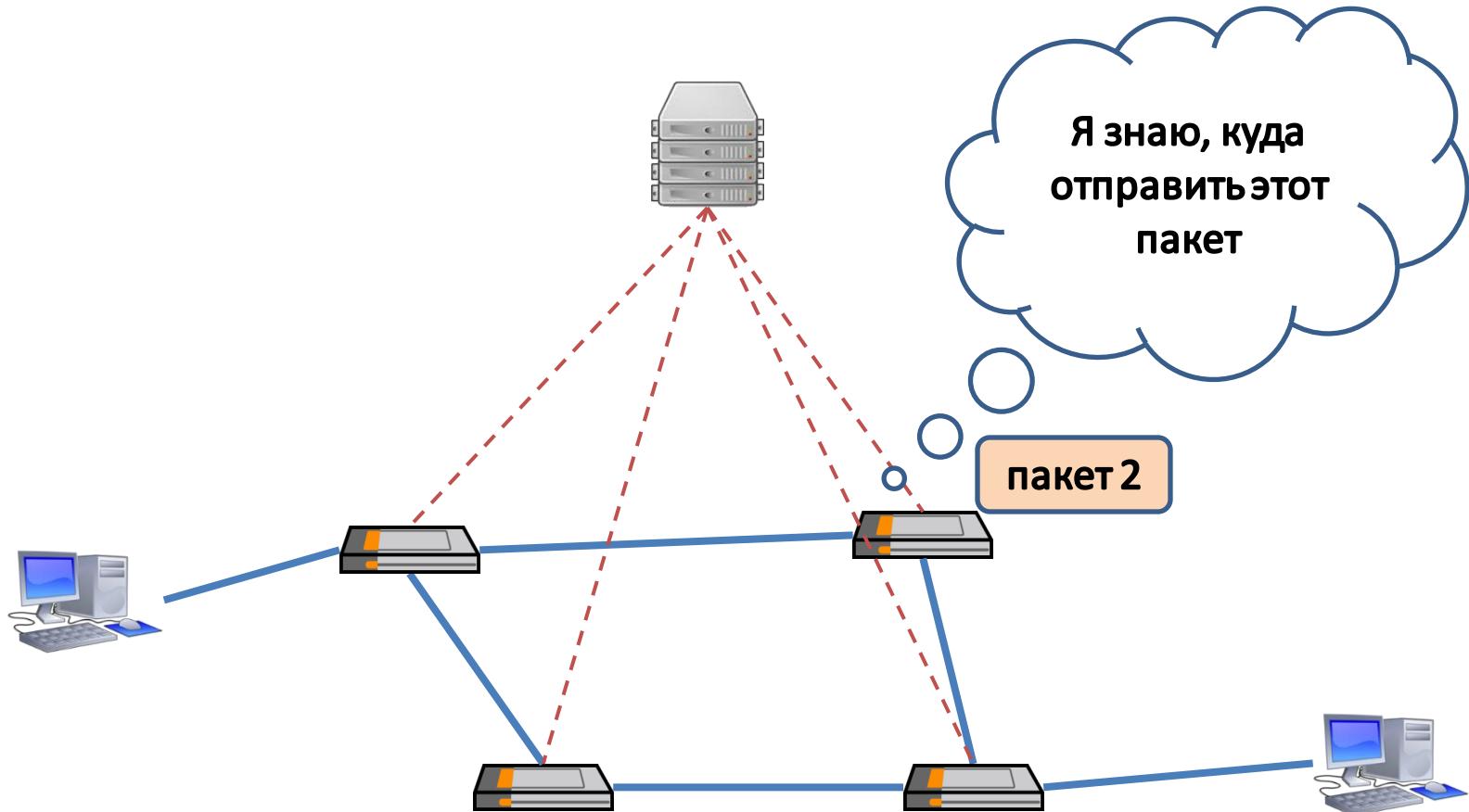
Чтобы пакет дошёл до адресата, нужно проложить для него маршрут через коммутаторы



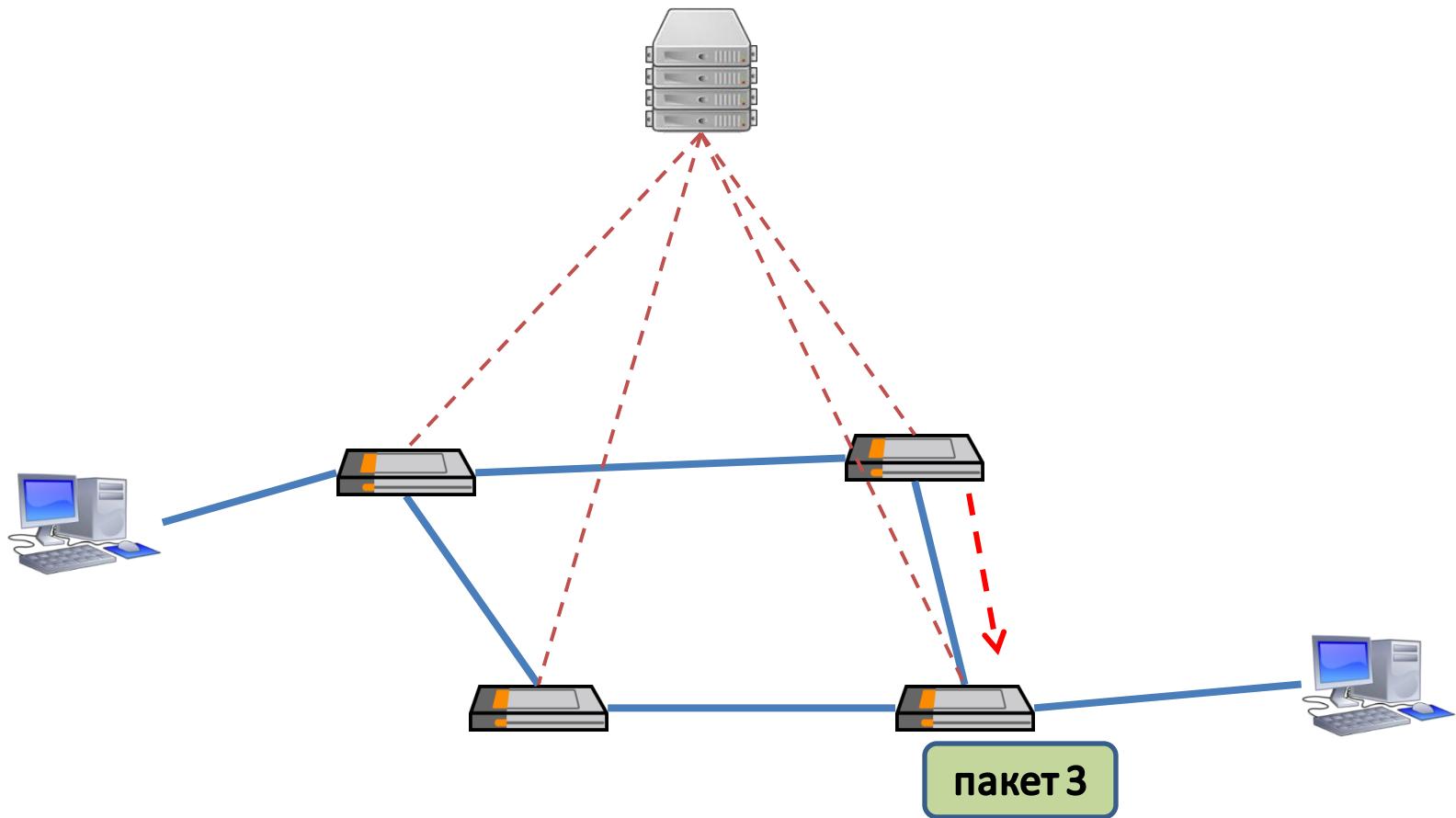
# Программно-конфигурируемые сети (ПКС)



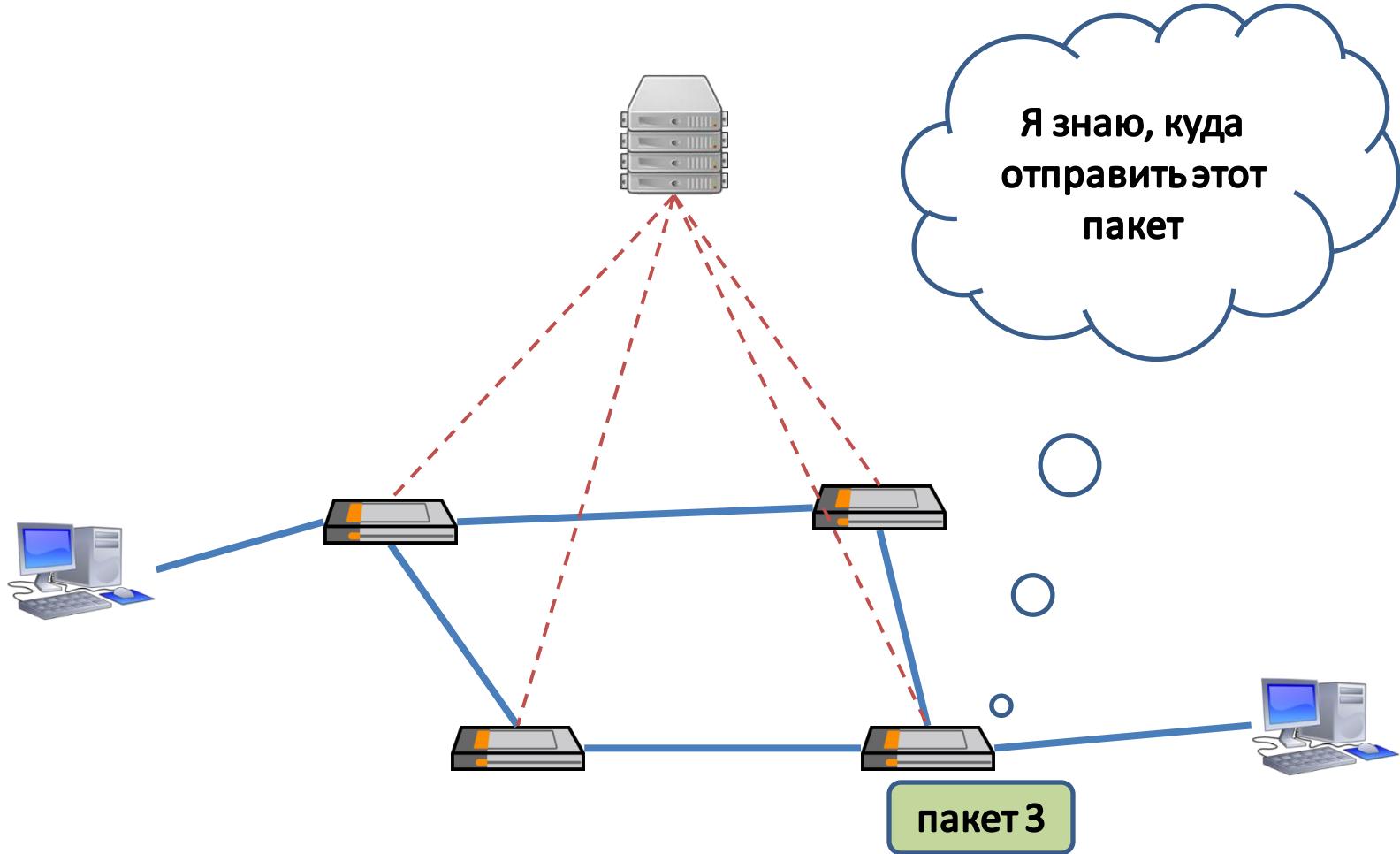
# Программно-конфигурируемые сети (ПКС)



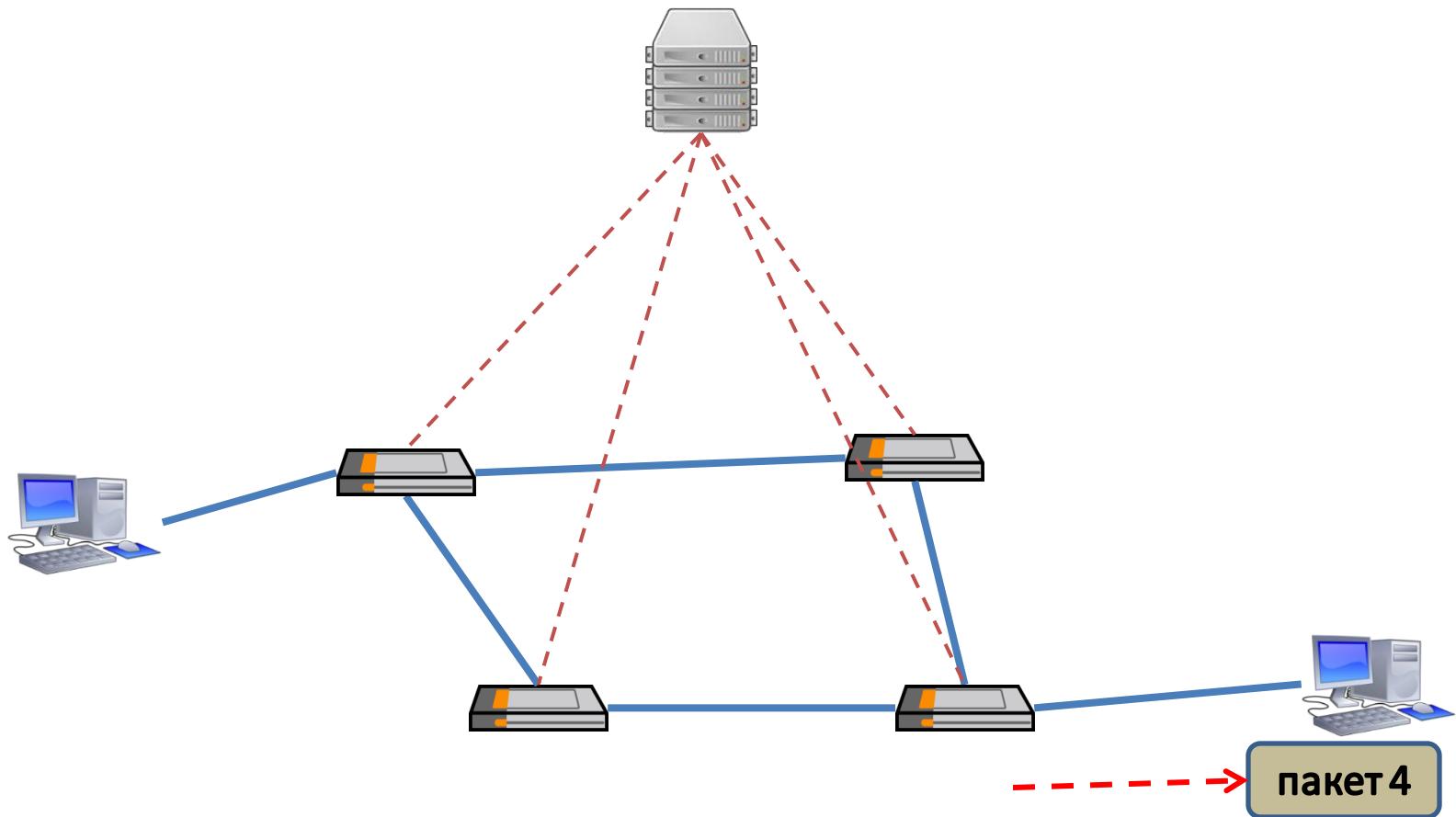
# Программно-конфигурируемые сети (ПКС)



# Программно-конфигурируемые сети (ПКС)



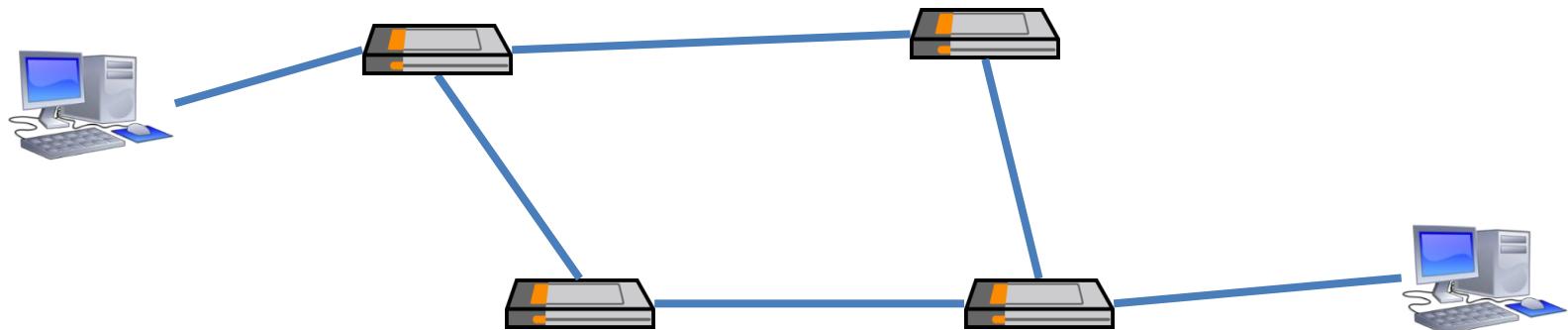
# Программно-конфигурируемые сети (ПКС)



# Требования к поведению сети

---

Как должна вести себя правильно настроенная сеть?

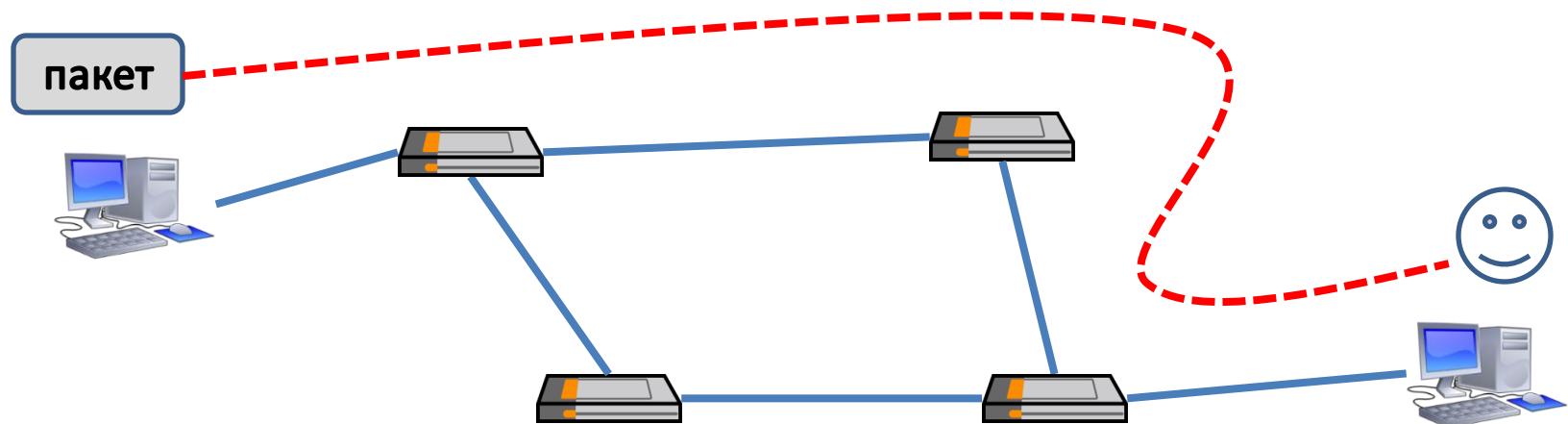


# Требования к поведению сети

Как должна вести себя правильно настроенная сеть?

Например:

**Пакет, отправленный в сеть, должен доходить до адресата**

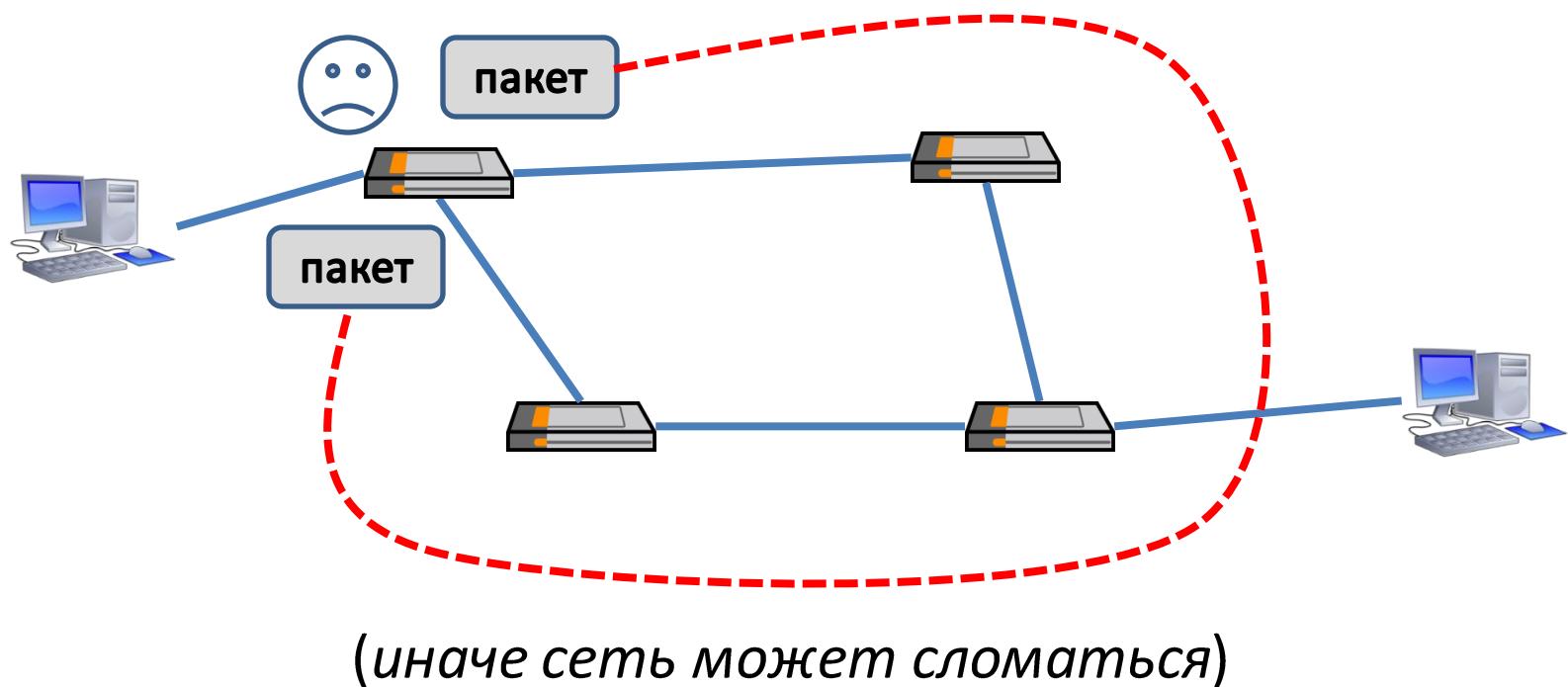


# Требования к поведению сети

Как должна вести себя правильно настроенная сеть?

Например:

**Пакет не должен дважды появляться на коммутаторе**

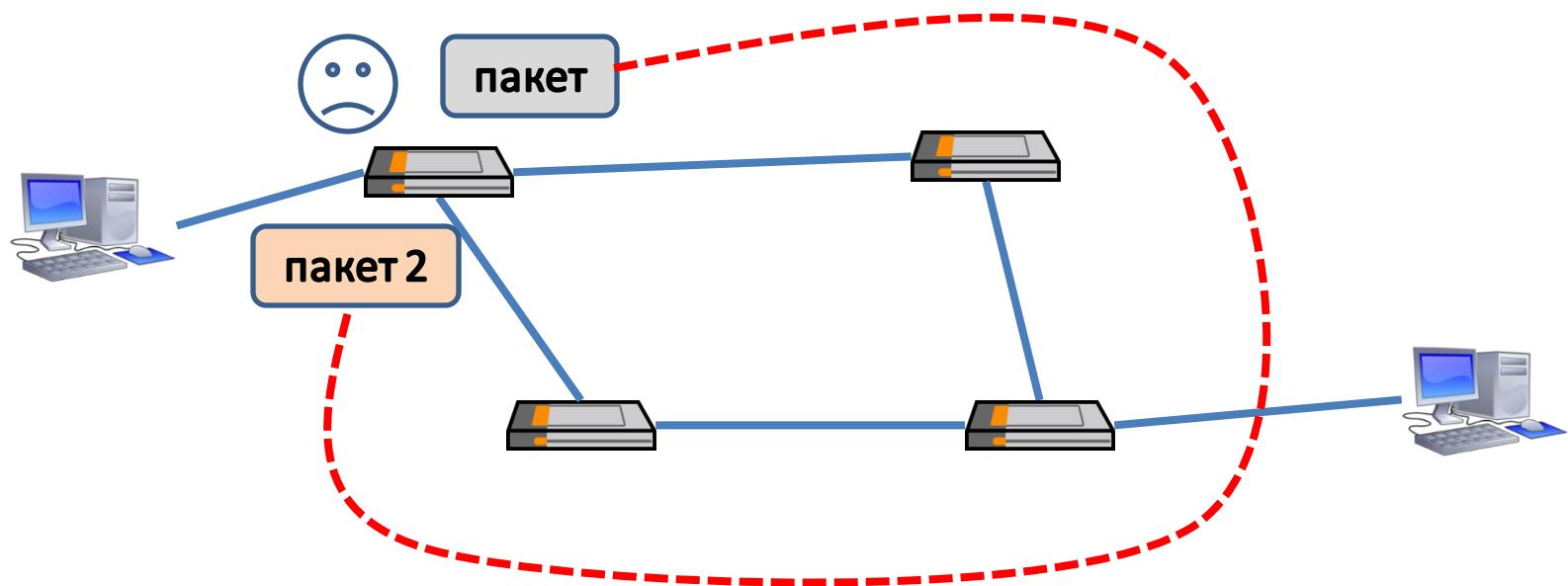


# Требования к поведению сети

Как должна вести себя правильно настроенная сеть?

Например:

**Пакет не должен дважды появляться на коммутаторе, даже если он изменился, двигаясь по узлам сети**



*(тогда сеть можно настроить лучше)*

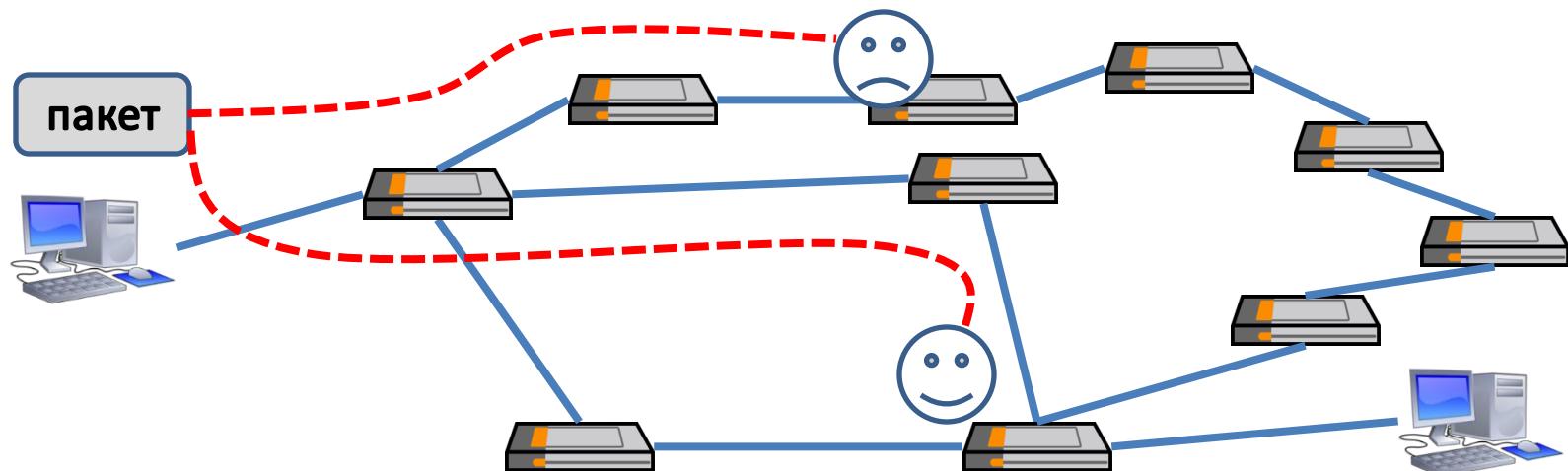
# Требования к поведению сети

Как должна вести себя правильно настроенная сеть?

Например:

## Маршрут движения пакета

**не должен быть неоправданно длинным**



*(время доставки пакета до адресата –  
важный критерий качества сети)*

# Верификация ПКС

---

**Верификация сети** – это строгая проверка того, что сеть отвечает предъявляемым к ней требованиям

# Верификация ПКС

---

**Верификация сети** – это строгая проверка того, что сеть отвечает предъявляемым к ней требованиям

Чтобы применить математические методы для верификации сети, необходимо

- построить математическую модель сети
- сформулировать требования к сети на математическом языке
- описать алгоритм проверки модели на соответствие требованиям

# Верификация ПКС

---

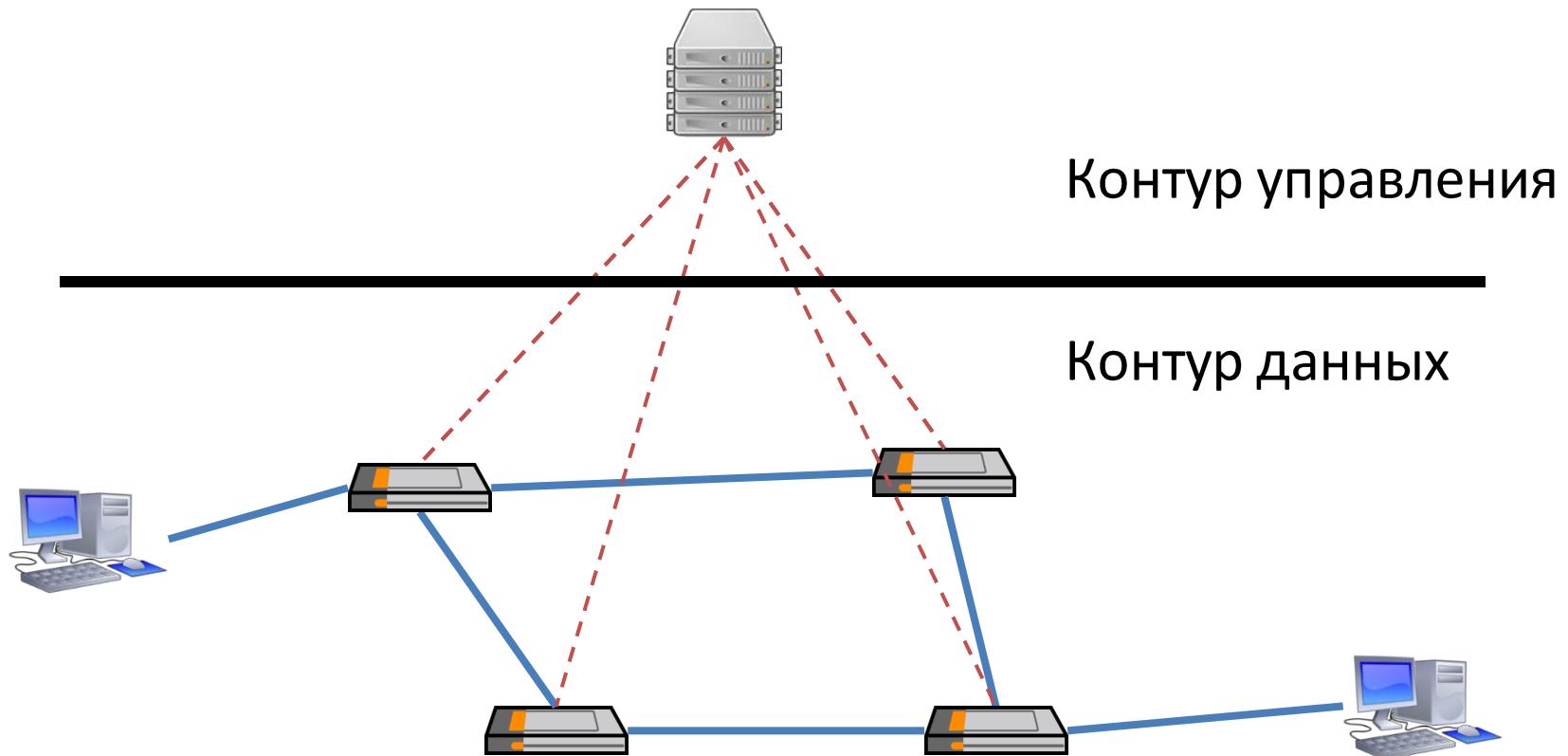
**Верификация сети** – это строгая проверка того, что сеть отвечает предъявляемым к ней требованиям

Чтобы применить математические методы для верификации сети, необходимо

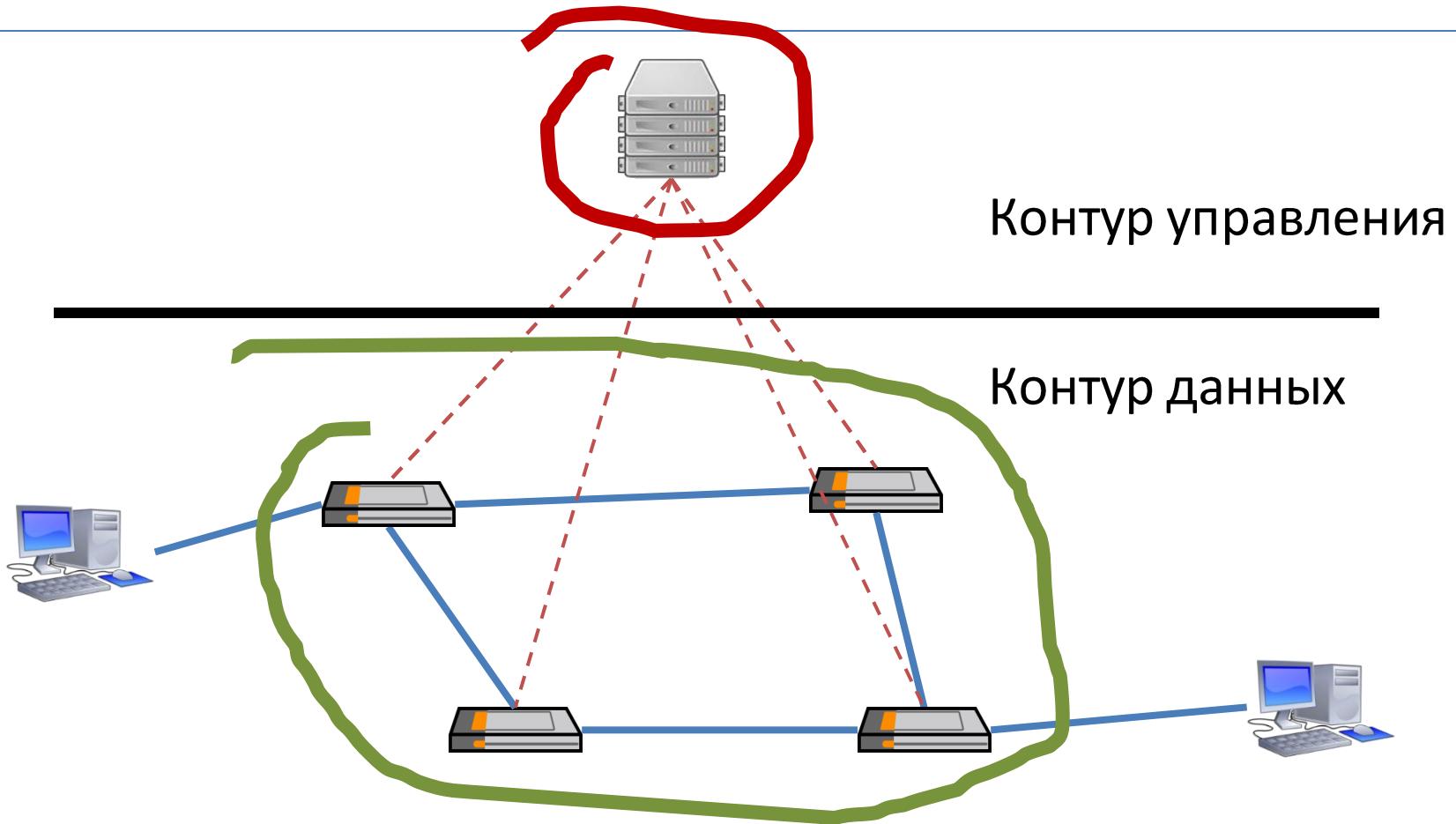
- построить математическую модель сети
- сформулировать требования к сети на математическом языке
- описать алгоритм проверки модели на соответствие требованиям

*Как построить модель  
программно-конфигурируемой сети?*

# Верификация ПКС



# Верификация ПКС



Достаточно построить модель **контроллера** или **контура данных**

# Верификация ПКС

---

*Изменения в контуре данных*



*Обновление конфигурации коммутаторов*

Контроллер реагирует на события, происходящие в сети, так,  
как это записано в его **программе**

# Верификация ПКС

---

*Изменения в контуре данных*



*Обновление конфигурации коммутаторов*

Контроллер реагирует на события, происходящие в сети, так,  
как это записано в его **программе**

Значит, достаточно убедиться,  
что программа контроллера написана *правильно*, то есть  
**верифицировать программу контроллера**

# Верификация ПКС

---

*Изменения в контуре данных*



*Обновление конфигурации коммутаторов*

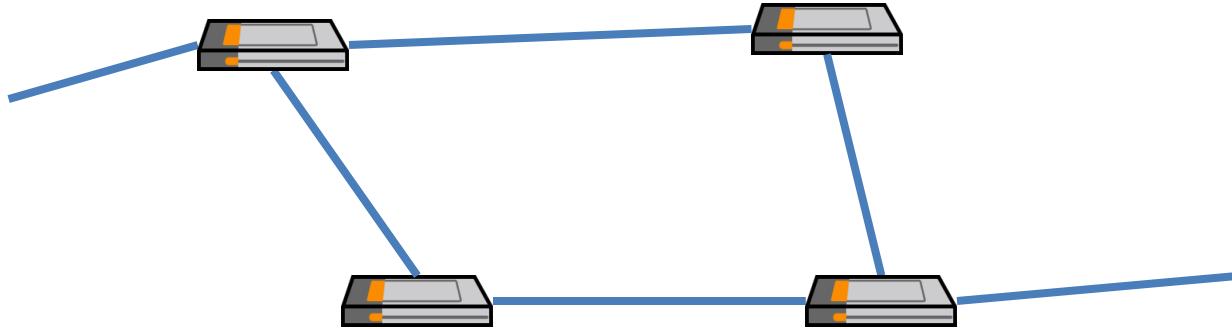
Контроллер реагирует на события, происходящие в сети, так,  
как это записано в его **программе**

Значит, достаточно убедиться,  
что программа контроллера написана *правильно*, то есть  
**верифицировать программу контроллера**

**Верификация программ** – это отдельное направление  
с огромным числом методов и результатов  
(и этим направлением мы занимаемся, но не для ПКС-контроллеров)

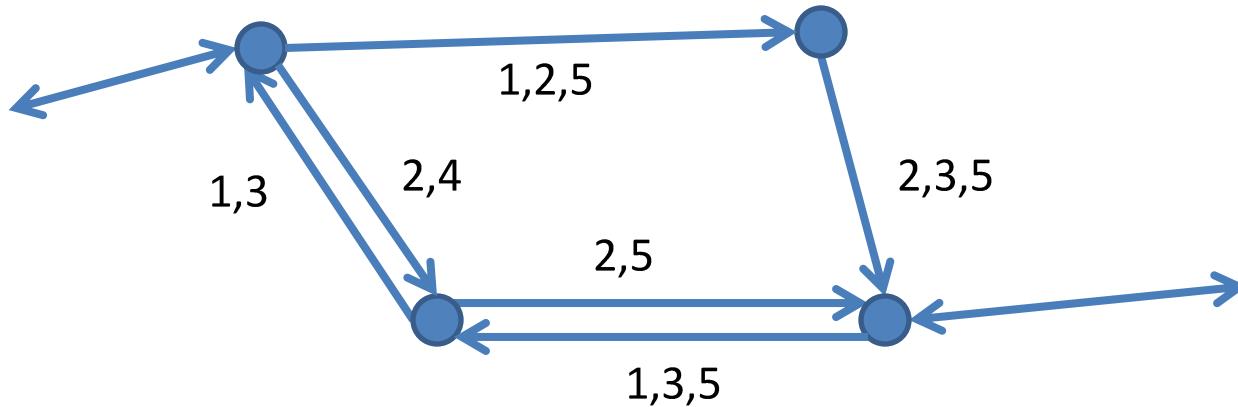
# Верификация ПКС

---



*А как может выглядеть модель контура данных?*

# Верификация ПКС

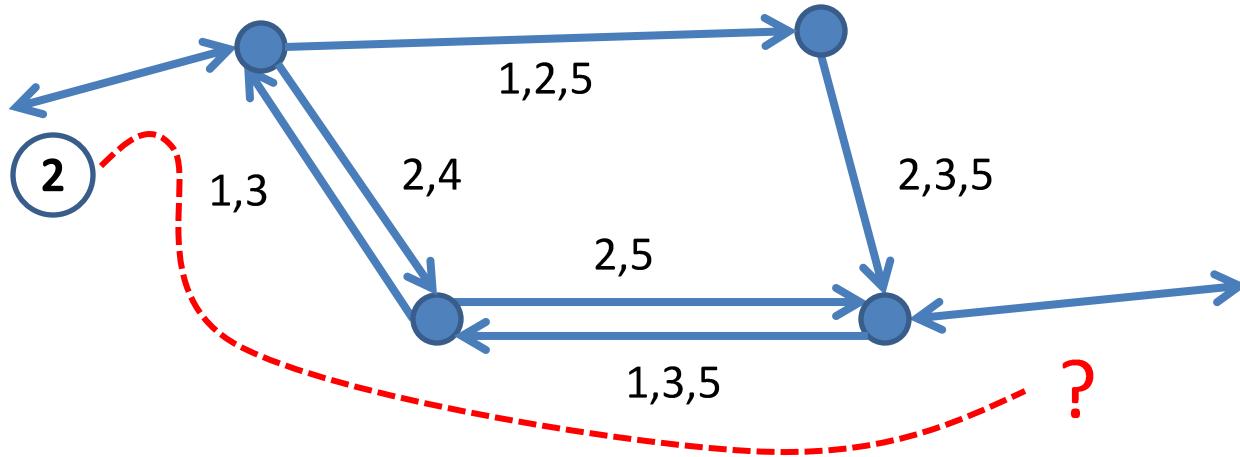


*А как может выглядеть модель контура данных?*

Например, можно описать контур данных **графом топологии сети** и пометить дуги видами пакетов, пересылающихся коммутаторами через соответствующие каналы связи

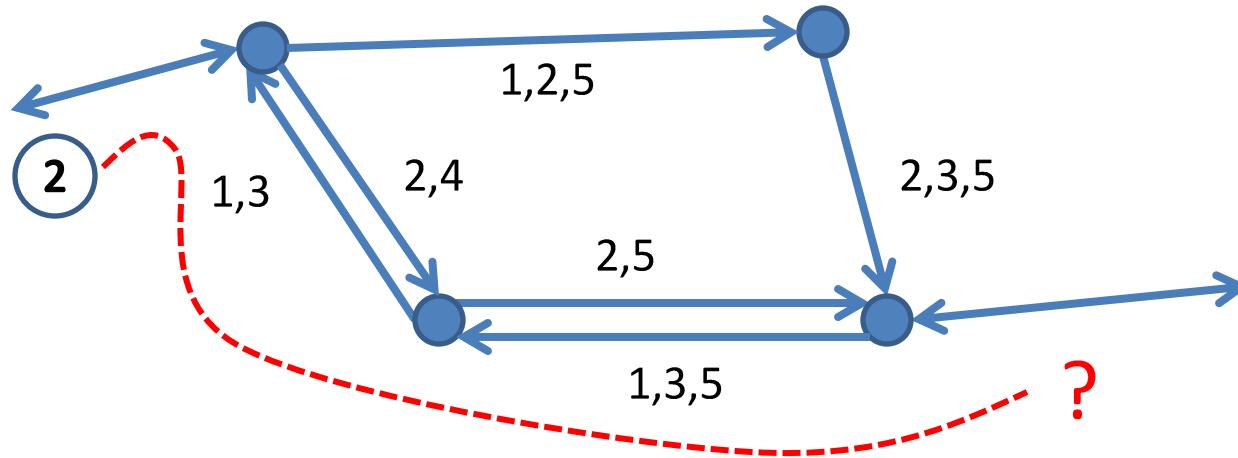
Тогда можно проверить любое требование к “слепку” сети в заданный момент времени, записанное в виде задачи на графике

# Верификация ПКС



Например, для обязательная доставка пакета адресату –  
это достижимость вершины в графе

# Верификация ПКС

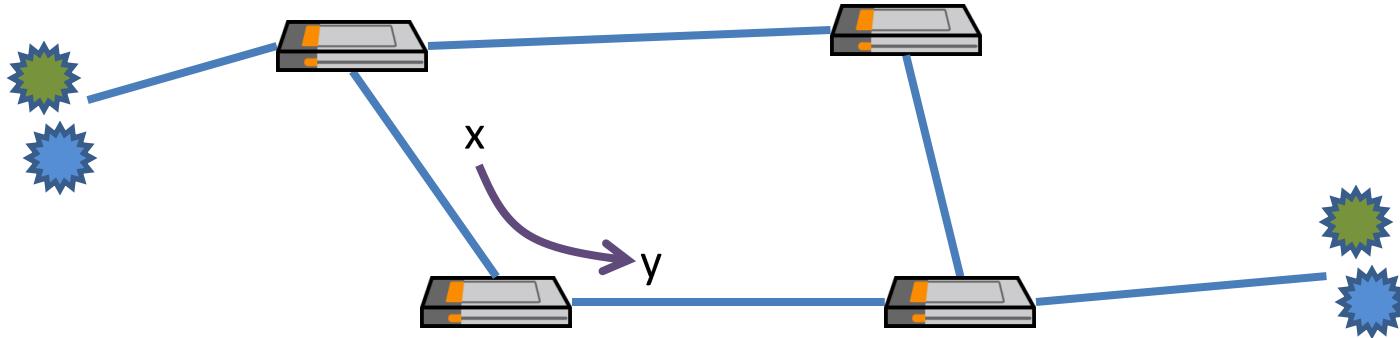


Например, для обязательная доставка пакета адресату – это достижимость вершины в графе

*Основной недостаток такого подхода:*

в сети может возникать огромное число различных видов пакетов, и чаще всего анализ графа сети оказывается неэффективным

# Верификация ПКС

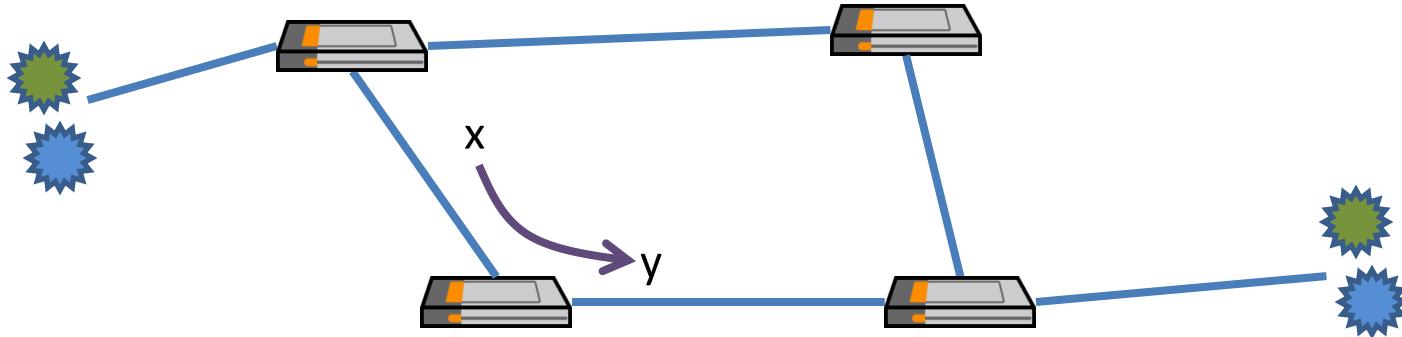


Есть и другой вид моделей контура данных: **реляционные модели**

Контур данных записывается в виде *набора отношений*,  
например:

- состояние пакета – это пара  $(p, h)$ , где
  - $p$  – положение пакета в сети
  - $h$  – вид пакета
- множество состояний входа в сеть:  $\text{In}(x)$
- множество состояний выхода из сети:  $\text{Out}(x)$
- отношение перенаправления пакетов:  $R(x,y)$

# Верификация ПКС



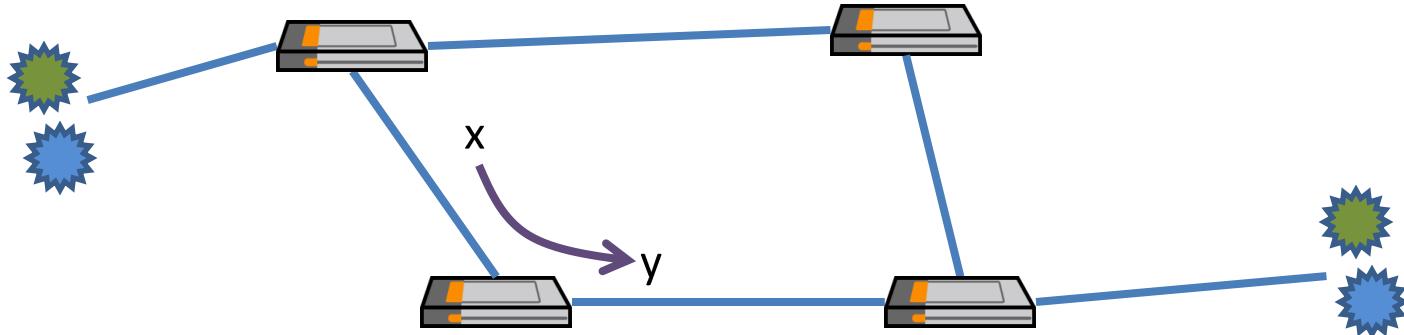
Отношение на множестве  $A$  можно расценивать как булеву функцию  $C(A) \rightarrow \{0,1\}$ ,  
где  $C(A)$  – множество двоичных кодов  $A$

Тогда модель контура данных – это **набор булевых функций**,  
и проверка требований, предъявляемых к сети, становится  
проверкой свойств булевых функций

При этом существуют способы хранения  
и работы с булевыми функциями, эффективные на практике

# Верификация ПКС

---

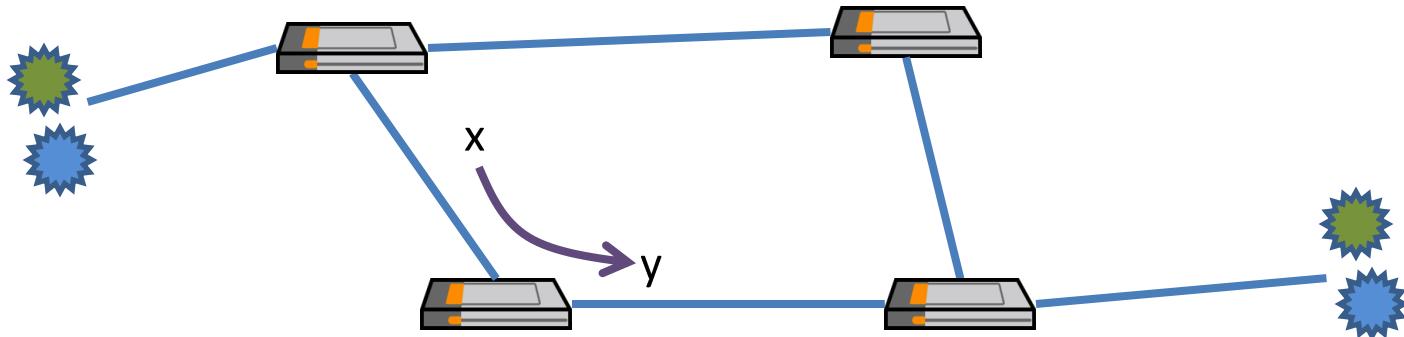


**Но проблемы все равно остаются:**

- конфигурация сети постоянно изменяется
- проверка требований к слепку сети должна быть настолько быстрой, чтобы результаты проверки всегда были актуальны

# Верификация ПКС

---

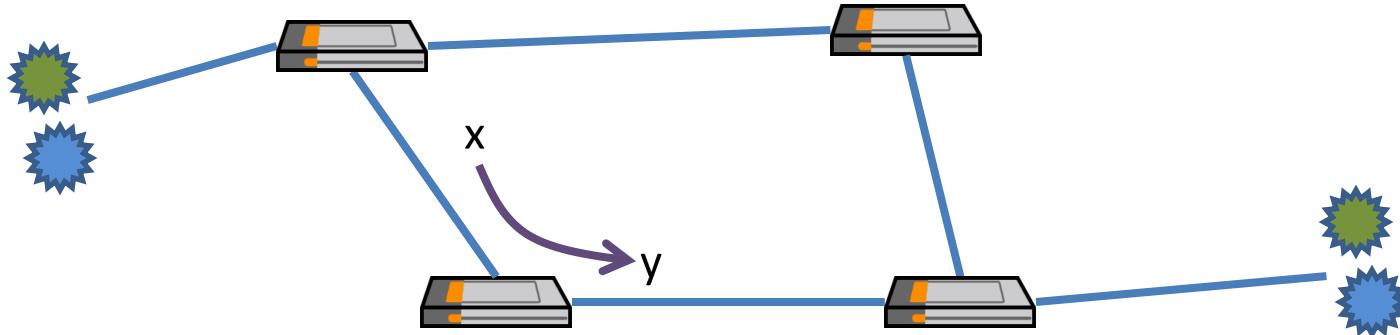


**Но проблемы все равно остаются:**

- конфигурация сети постоянно изменяется
- проверка требований к слепку сети должна быть настолько быстрой, чтобы результаты проверки всегда были актуальны

*А насколько это быстро?*

# Верификация ПКС



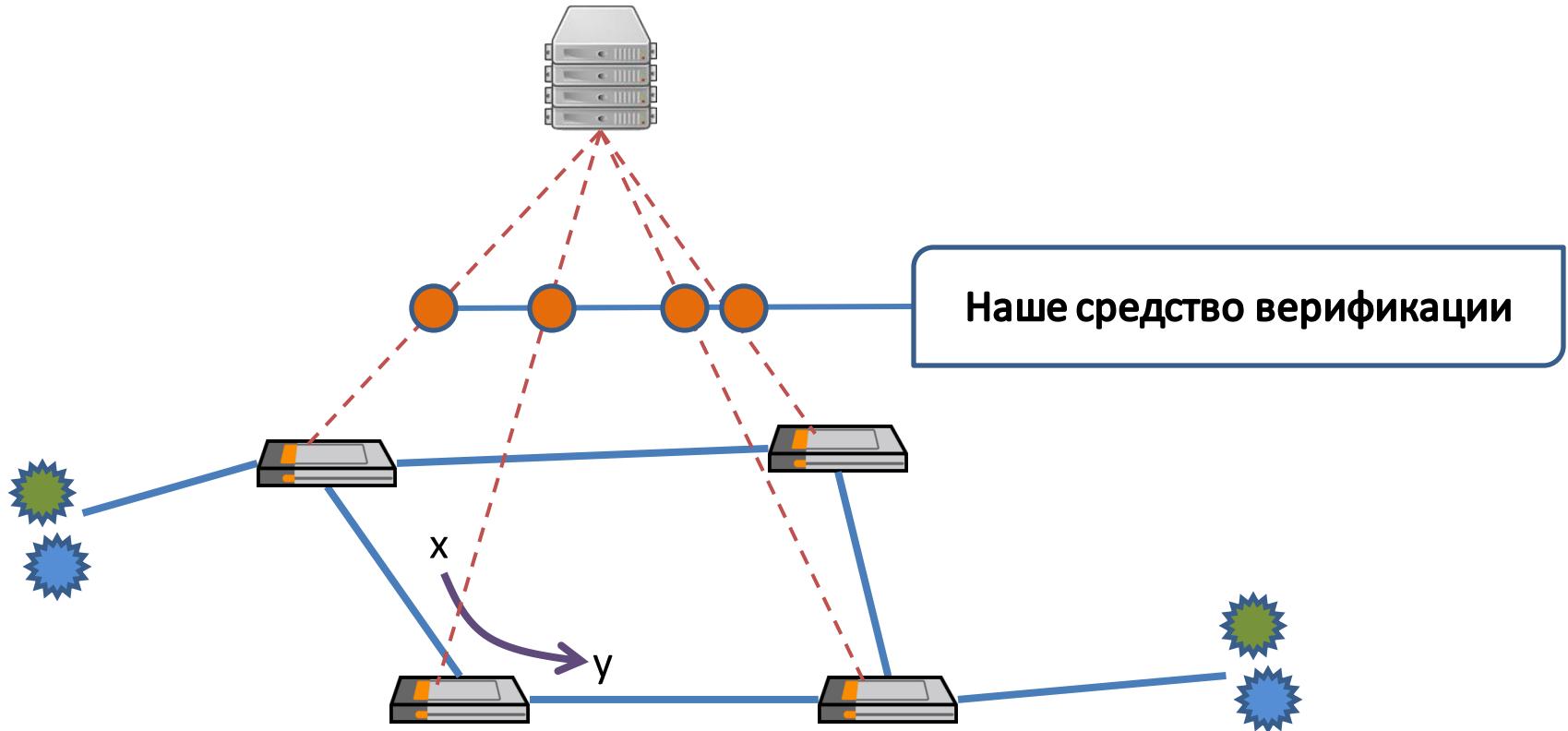
**Но проблемы все равно остаются:**

- конфигурация сети постоянно изменяется
- проверка требований к слепку сети должна быть настолько быстрой, чтобы результаты проверки всегда были актуальны

*А насколько это быстро?*

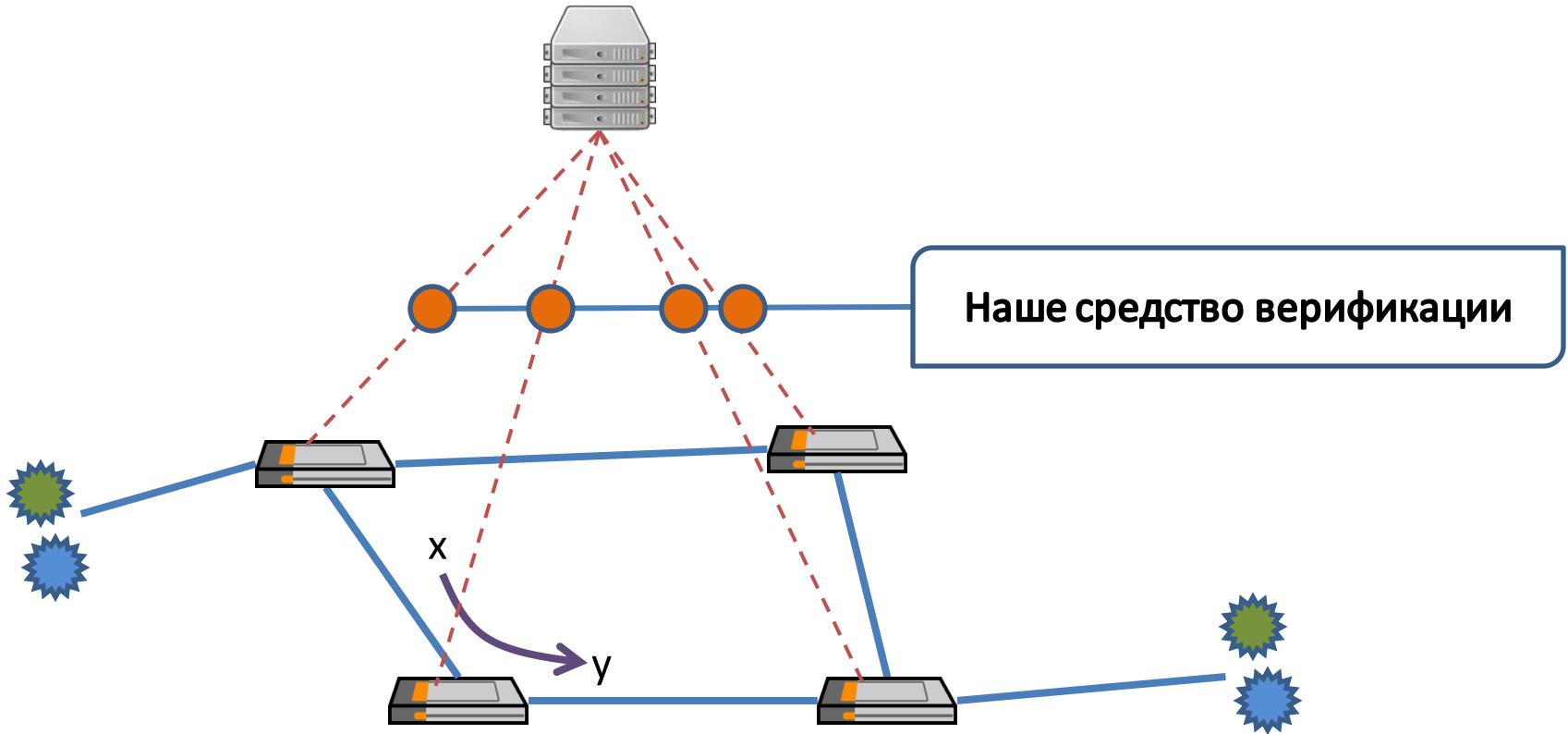
Сейчас средство верификации считается неплохим, если оно способно проверять требования за **миллисекунды**, и хорошим – за **микросекунды**

# Что делаем мы



Мы придумали и реализовали реляционную модель и алгоритмы верификации контура данных ПКС, работающие достаточно хорошо для не очень больших сетей и для широкого класса требований (в том числе *всех требований, которые были в начале рассказа*)

# Что делаем мы



Но (как обычно) этого недостаточно,  
и работа в этом направлении продолжается