

# РЕАЛИЗАЦИЯ УСТАНОВЛЕНИЯ СОСЕДСКИХ ОТНОШЕНИЙ В ПРОТОКОЛЕ OSPFv3

*Е.А. Черкашин, ИДСТУ СО РАН  
Д.А. Паккерт, ИрНИТУ*

*Иркутск*

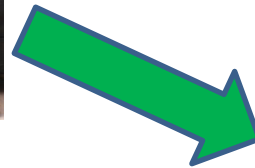
# Переход к интернету вещей



2011 год

## Необходимость IPv6

## Необходимость динамической маршрутизации

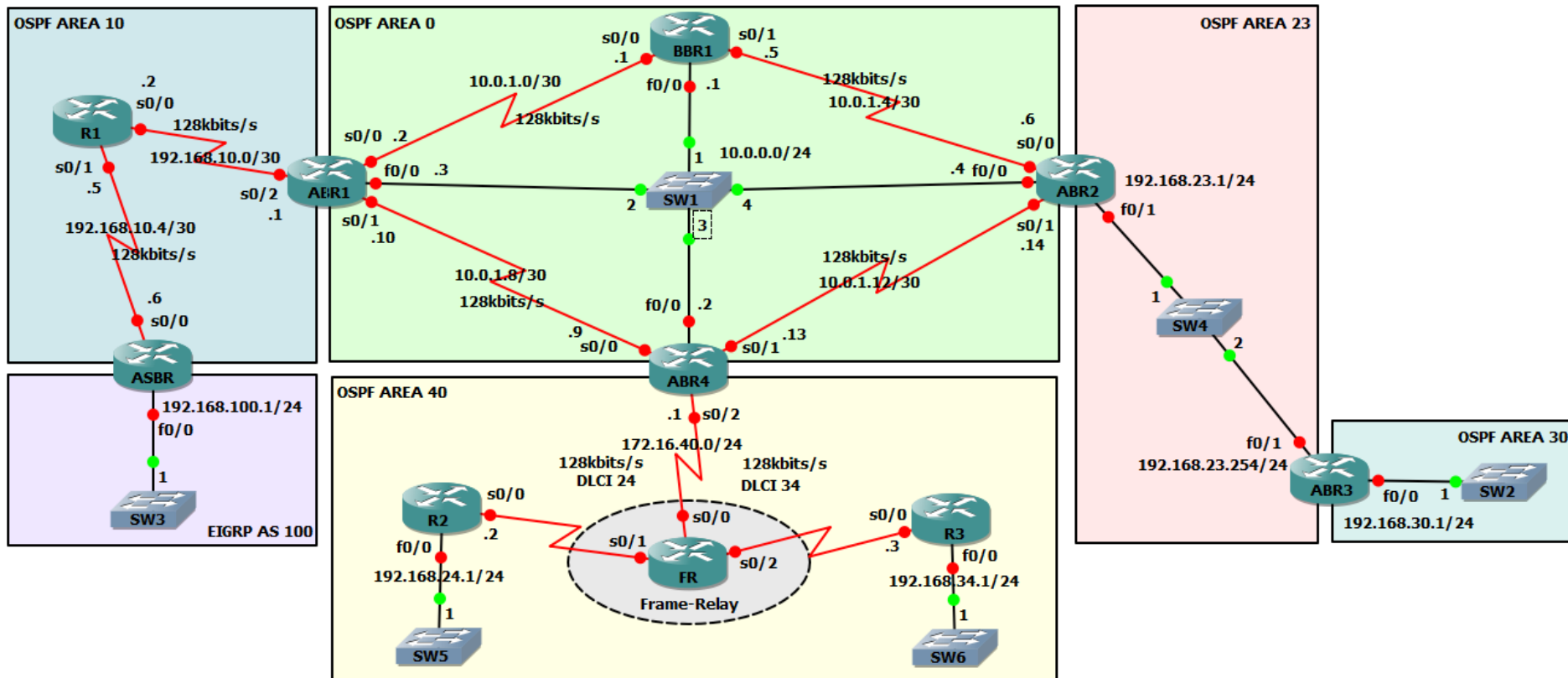


SDN – программно-определяемые сети  
Сети, которые являются основой

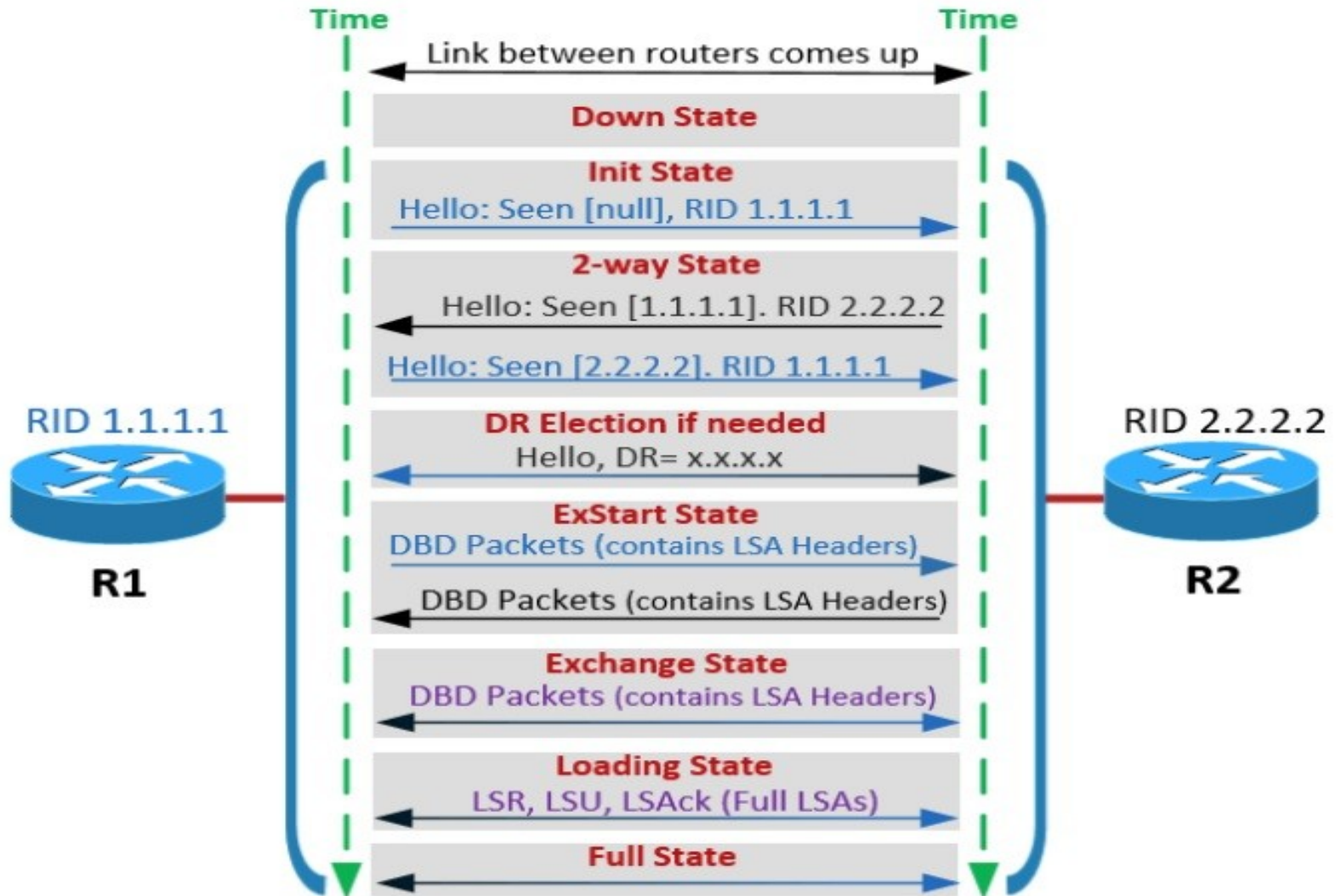
- виртуальных сетей предприятий;
- облачных технологий.



# Выбор кратчайшего пути по наименьшей стоимости. Алгоритм Э. Дейкстры



# Состояния маршрутизаторов во время установления соседских отношений



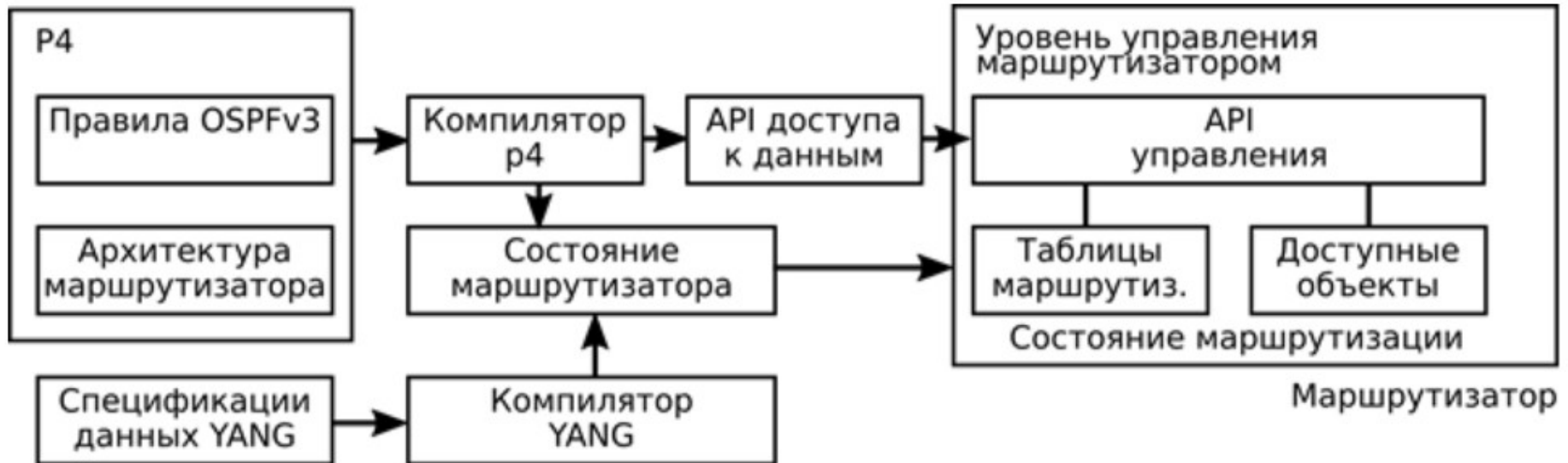
# Цели и задачи исследования

**Цель исследования** — реализовать OSPFv3 при помощи декларативных языков P4 и YANG, при этом на традиционном для программирования демонов маршрутизации языке C разрабатываются только алгоритмы на графах и взаимодействие с операционной системой.

**Задачи:**

1. Представление уровня управления SDN/Маршрутизатором;
2. Анализ существующих спецификаций YANG для протокола OSPFv3;
3. Реализация уровней L2, L3 в среде P4;
4. Реализация алгоритмов и интерфейса с уровнями L2, L3 к алгоритмам и операционной системе.

# Синтез структур логического уровня управления маршрутизатором



# YANG-описания структур данных для хранения состояний маршрутизатора

```
grouping neighbor-state {  
  description  
    "OSPF neighbor state."  
  leaf address {  
    type inet:ip-address;  
    config false;  
    description "Neig. IP";  
  }  
}
```

```
leaf dr-router-id {  
  type rt-types:router-id;  
  config false; }  
leaf dr-ip-addr {  
  type inet:ip-address;  
  config false;  
  description "Neig's DR IP."  
} ... }
```

# Обработка пакетов: правила R4

```
control_neighbor_state(...h, ...inp,
...err, ...out, ...rou) {
    table down_hello {
        key = {h.ospf.nei.id==none;
            rou.st == DOWN;}
        actions = {set_hello;}}
    table init_hello {
        key = {h.ospf.nei.id!=none;
            rou.st == DOWN}
        actions = {set_hello;
            set_nrID;}}
```

```
    action set_hello{
        out.ospf.my.id =
            API.get_my_id();}
    action set_nrID {
        out.ospf.nei.id=rou.nei.id;
        rou.st = INIT;}
    apply {
        down_hello.apply();
        init_hello.apply();
        ... } ...
}
```



# Заключение

- Представлены основные идеи подхода к реализации;
- Показаны примеры спецификаций структур данных и части автомата, моделирующего состояние маршрутизатора и его соседей, а также
- Архитектура процесса проектирования средств ДМ.

Следующей задачей, решаемой в данном проекте, является адаптация компилятора P4 к API алгоритмов вычисления кратчайших расстояний Э.Дейкстры.