



# Summer Camp Software Competition 2019

---

王清和

August • 4 • 2019

## Agenda

1 思路

2 数据输入

3 任务决策  
(当前)

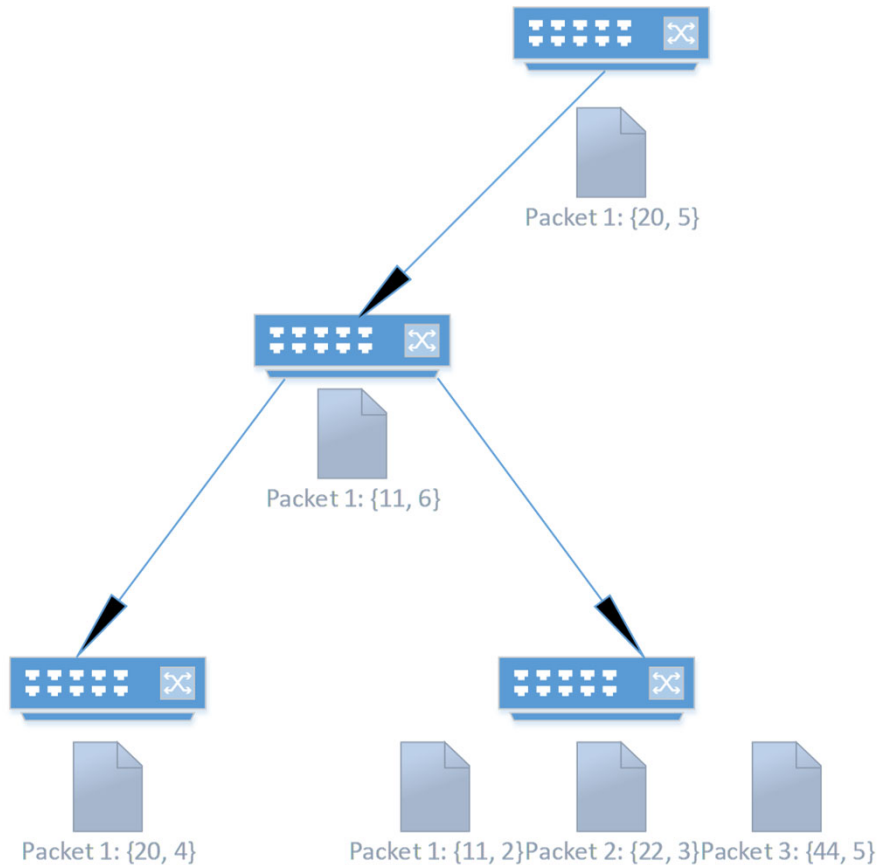
4 任务进度  
(当前)

5 示例

6 后期改进

# 思路

如何用尽可能少的时间进行路由器集群的固件升级。



```
int main()
{
    Input_Init();    //输入数据初始化
    start = clock(); //开始计时
    while(1)
    {
        //循环包，判断、决策
        Make_Decision();

        //检查时间，判断有没有包下载/更新完成
        check_time();
    }
    return 0;
}
```

## 数据输入

`struct Router Router_tree[500];` //定义路由器数组，一共Router\_Number\_N个路由器

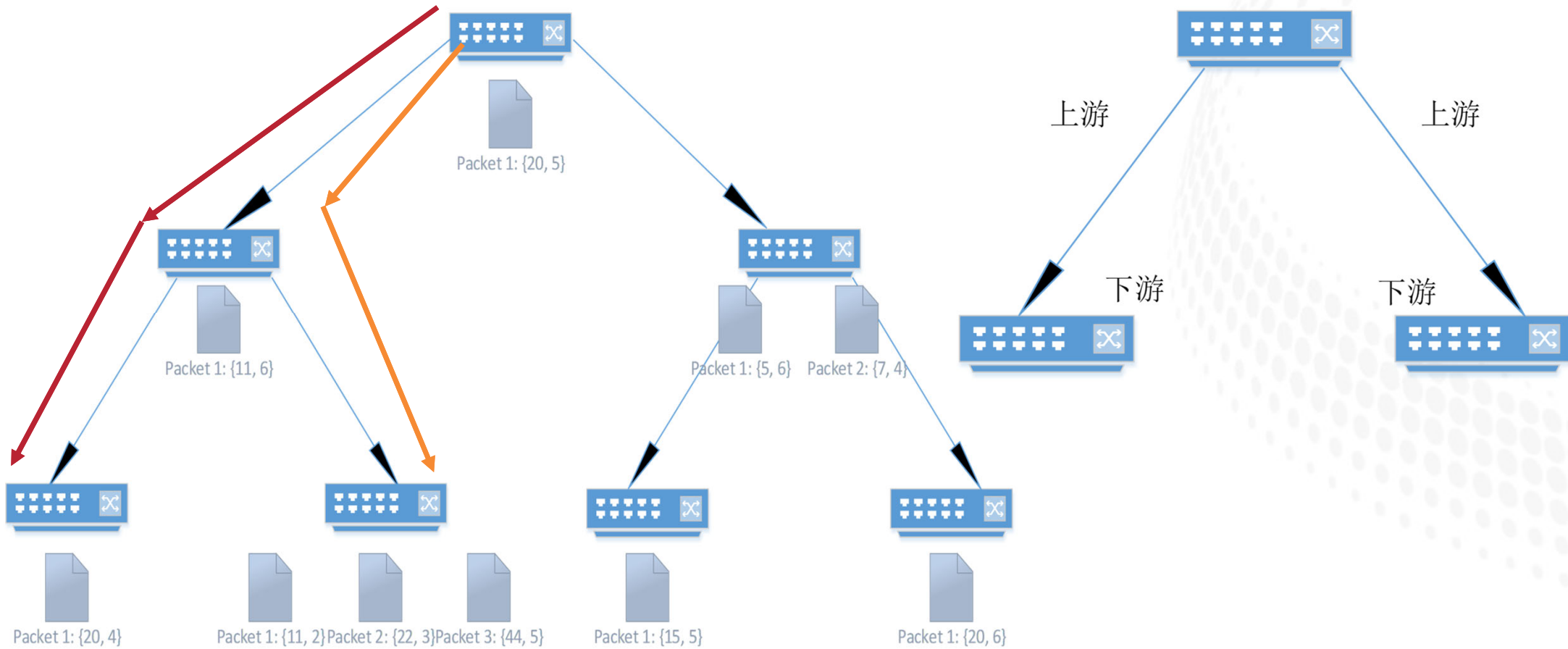
限制条件：假设有N（N小于500）个路由器，编号1到N。

`struct Package Packages_ULimit[5000];` //最多有500\*10个包

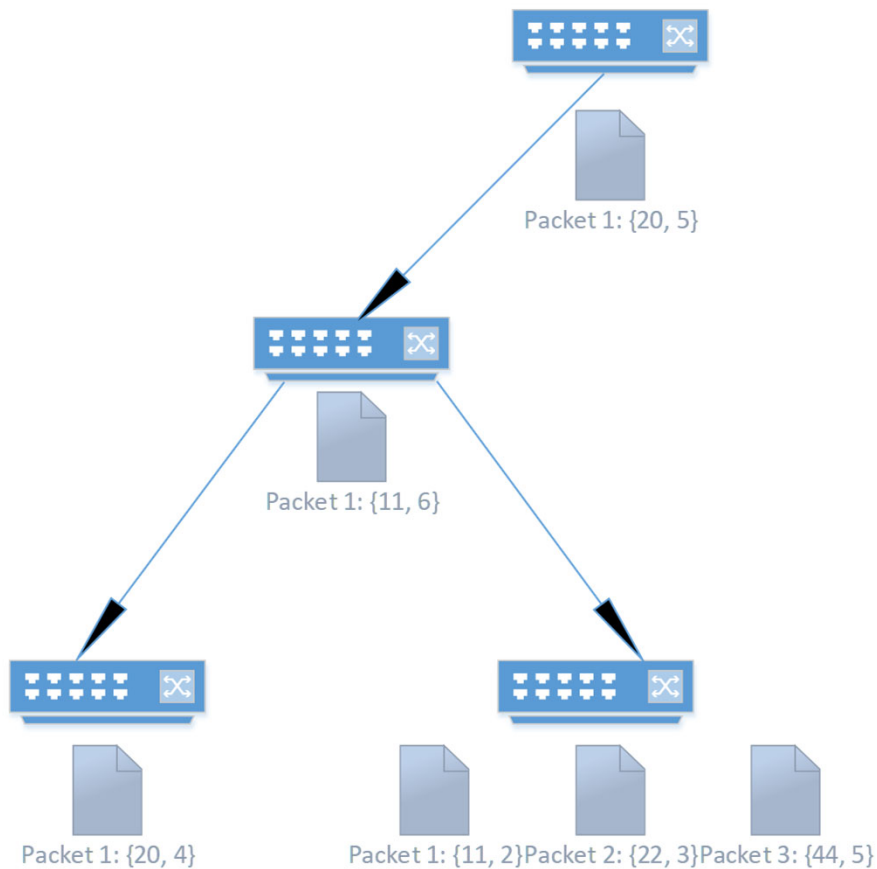
限制条件：单个路由器升级补丁包个数不超过10。

.....（就不一一展开介绍了）

# 数据输入



## 数据输入



3个影响因子（主权重）：

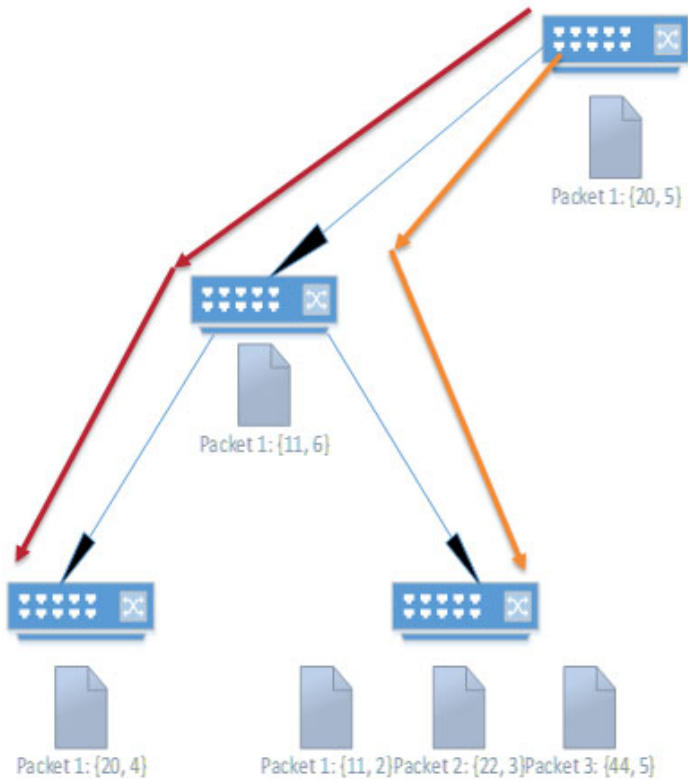
1. 层级（占比0.8）
2. 下载时间（占比0.1）
3. 更新时间（占比0.1）

数据归一化（以层级为例）：

$$level' = \frac{(level - level_{min})}{(level_{max} - level_{min})}$$

得分 = 0.8\*层级' + 0.1\*下载时间' + 0.1\*更新时间'

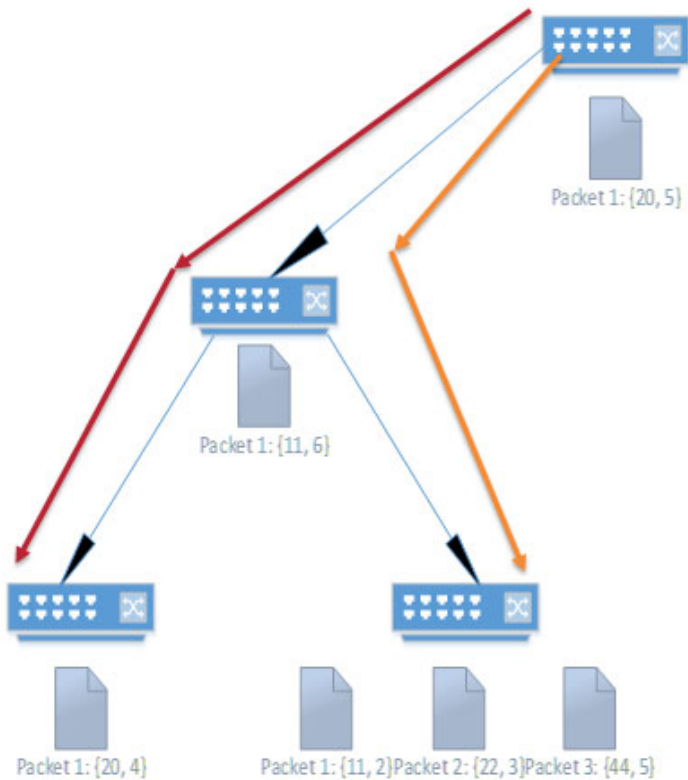
## 任务决策（当前包）



当前包**下载**需满足的条件：

- 当前包没下载过
- 当前路由器在空闲状态
- 限制条件：同一支路同时下载升级包的路由器数目不能超过上限P
- 上级路由器此时没在更新

## 任务决策（当前包）



当前包**更新**需满足的条件：

- 当前包已经下载好了
- 当前路由器在等待更新状态
- 下级路由器此时没在下载
- 限制条件：同时更新路由器数目上限为Q  
(**solution**: 判断此时所有正在更新路由器数目)
- 下级路由器都已经更新完毕
- 下级在更新时，什么情况我可以一起更新？

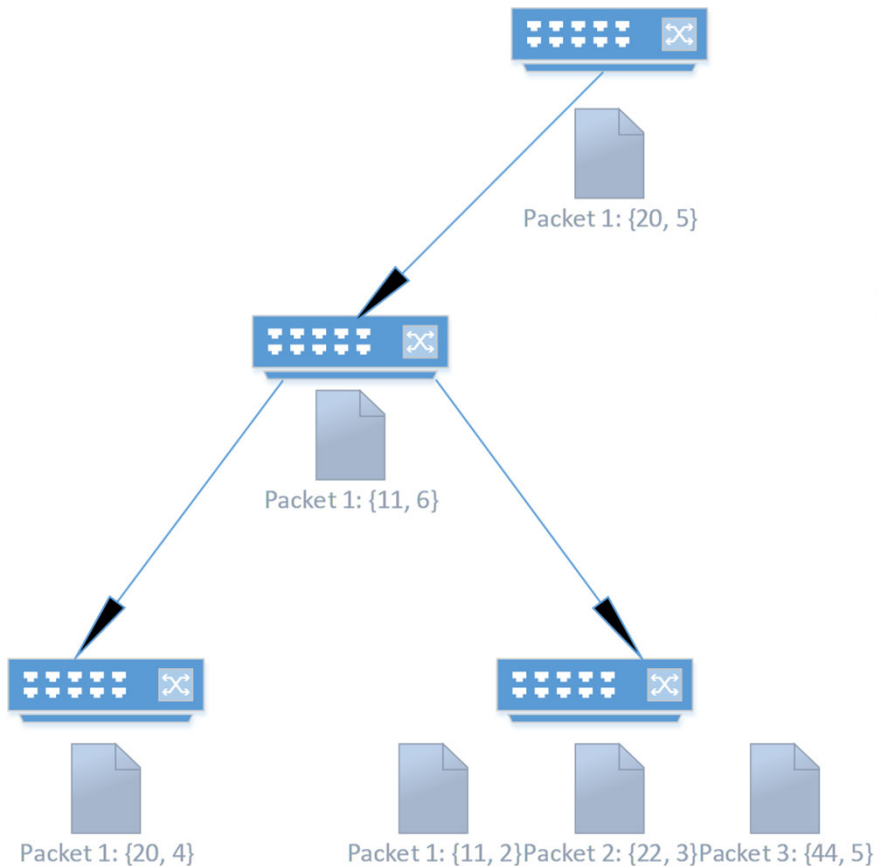


## 任务决策（当前包）

下级在更新时，什么情况我可以一起更新？

满足：

- 下级都在更新安装包，
- 本级路由器的更新时间小于等于下级所有路由器距离更新结束的时间



## 任务进度（当前）



`Current_Time = (clock()-start) / CLK_TCK;`

// 读取当前时间，用来判断每个包是否下载/更新完毕了

## 示例二

### 输入数据：

4 3 10 11

1 2

2 3

2 4

1 1 20 5

2 1 11 6

3 1 20 4

4 3 11 2 22 3 44 5

路由器数N，单向电线数M，同一支路同时接收升级包路由器上限P，同时更新电站上限Q

路由器到路由器的单向边

路由器信息：编号，升级补丁包个数，接收升级包耗时和更新耗时

### 输出数据：

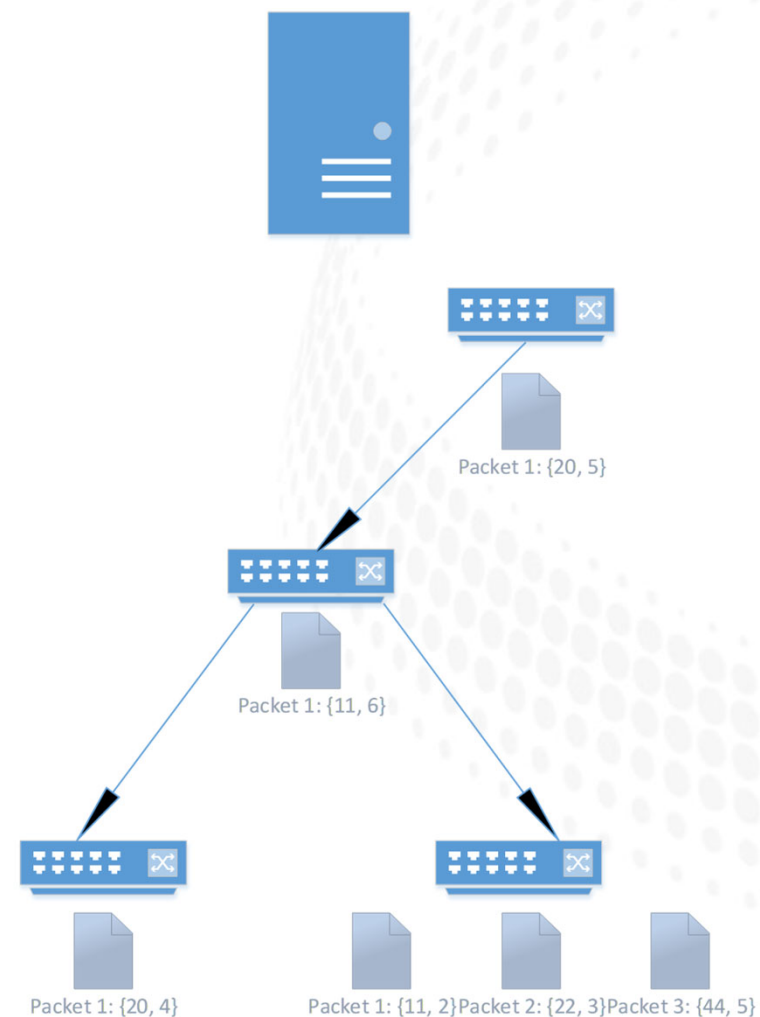
电站标识，接收开始时刻，更新开始时刻

1 0 82

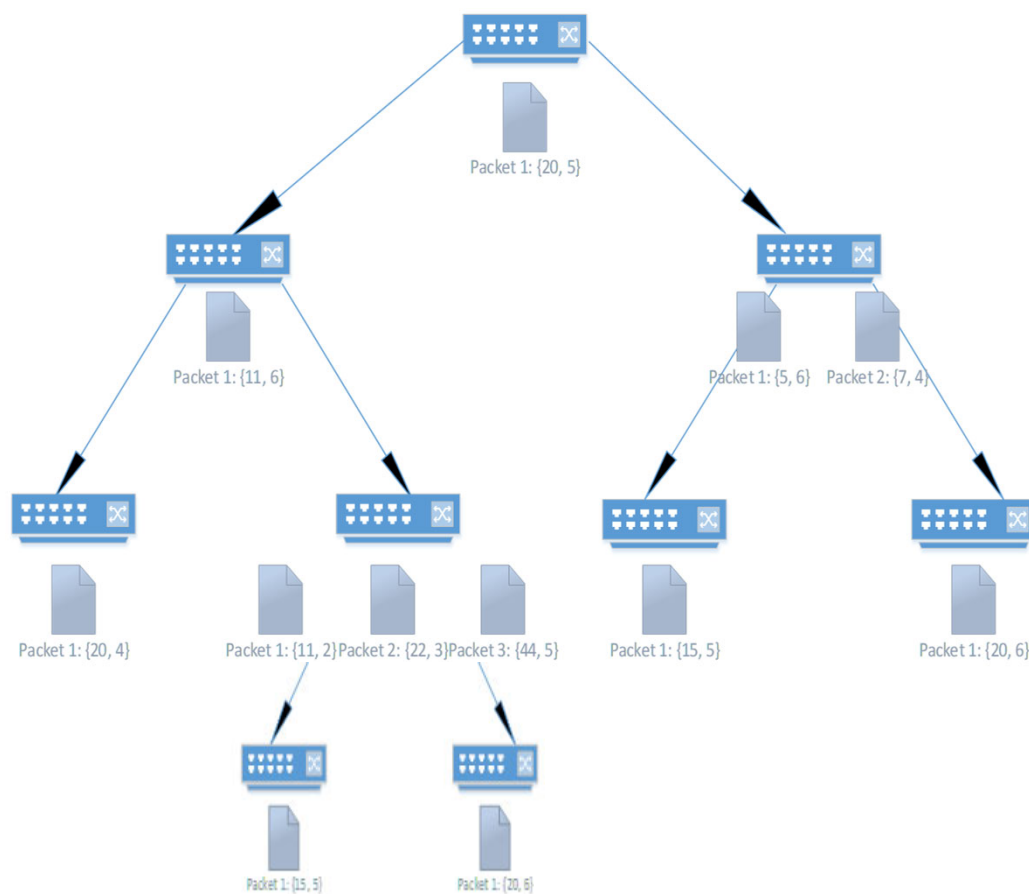
2 0 82

3 0 20

4 0 11 13 35 38 82 总耗时：88s



## 后期改进



限制条件：当路由器编号为5的倍数时，路由器在接收升级包结束之后，**自动更新**。

Bad solution: 编号为 $5N$ 的路由器，需要在所有下级路由器都更新完才开始下载。

Better solution: 只要不打断、不耽误下级路由器下载安装包，就可以下载。

## 后期改进

得分 = 0.8\*层级' + 0.1\*下载时间' + 0.1\*更新时间'

遍历打分的3个系数，遍历整个模拟过程多次，  
寻找时间最短的打分顺序



# Thank you

---



[www.rockwellautomation.com](http://www.rockwellautomation.com)