

# Reliable file transfer using Go-Back-N

# Implementing a Distance Vector Routing Protocol

## 计算机网络项目报告

讲者：崔晨曦

小组成员：崔晨曦 孙一钧 宋启帆 张国安 宋学长 邹欣然

北京理工大学计算机学院

2021年6月9日



**北京理工大学**  
BEIJING INSTITUTE OF TECHNOLOGY

# Outline

- Reliable file transfer using Go-Back-N
  - 项目简介
  - 设计思路
  - 程序演示
  - 结果分析
- Implementing a Distance Vector Routing Protocol
  - 项目简介
  - 设计思路
  - 程序演示
  - 结果分析

# Reliable file transfer using Go-Back-N

# 项目简介 (Go-Back-N)

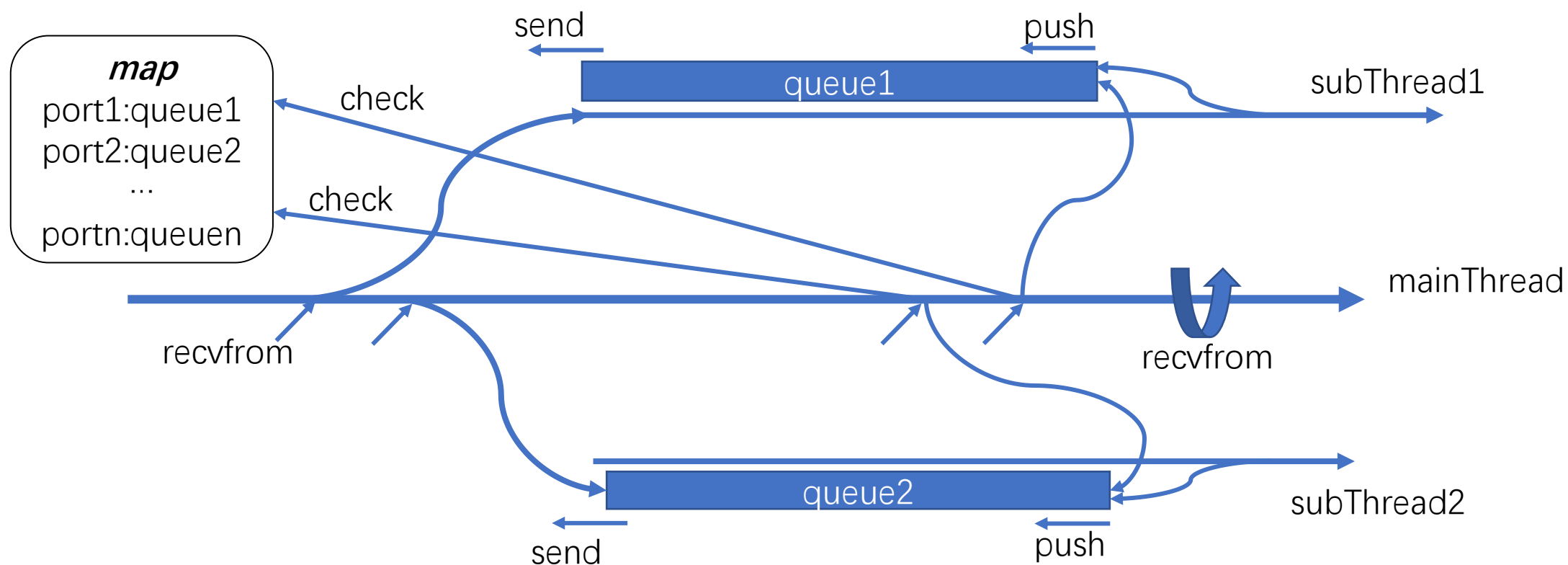
- 实现采用GBN协议的可靠文件传输
- 基于教材p182协议5
- 实现多个host同时传输
- 采用多线程、队列等技术

# 设计思路 (Go-Back-N)

- 帧结构
  - 帧序号
  - 数据
  - ack
  - crc校验
  - 源端口号

```
class Frame:
    def __init__(self, frame_nr, frame_expected, port):
        self.sender = port
        self.data = ''
        self.seq = frame_nr
        self.ack = (frame_expected + MAX_SEQ) % (MAX_SEQ + 1)
        self.checksum = ''
```

# 设计思路 (Go-Back-N)



# 程序演示 (Go-Back-N)

# 结果分析 (Go-Back-N)

```
309, send: pdu_to_send=2, status=New, ackedNo=4
310, send: pdu_to_send=3, status=New, ackedNo=4
311, send: pdu_to_send=4, status=New, ackedNo=4
283, receive: pdu_exp=5, pdu_rcv=5, status=OK
284, receive: pdu_exp=6, pdu_rcv=6, status=OK
285, receive: pdu_exp=0, pdu_rcv=0, status=OK
286, receive: pdu_exp=1, pdu_rcv=1, status=OK
287, receive: pdu_exp=2, pdu_rcv=2, status=OK
288, receive: pdu_exp=3, pdu_rcv=3, status=OK
312, send: pdu_to_send=5, status=New, ackedNo=3
313, send: pdu_to_send=6, status=New, ackedNo=3
314, send: pdu_to_send=0, status=New, ackedNo=3
315, send: pdu_to_send=1, status=New, ackedNo=3
316, send: pdu_to_send=3, status=TO, ackedNo=3
317, send: pdu_to_send=4, status=TO, ackedNo=3
318, send: pdu_to_send=5, status=TO, ackedNo=3
319, send: pdu_to_send=6, status=TO, ackedNo=3
320, send: pdu_to_send=0, status=TO, ackedNo=3
```

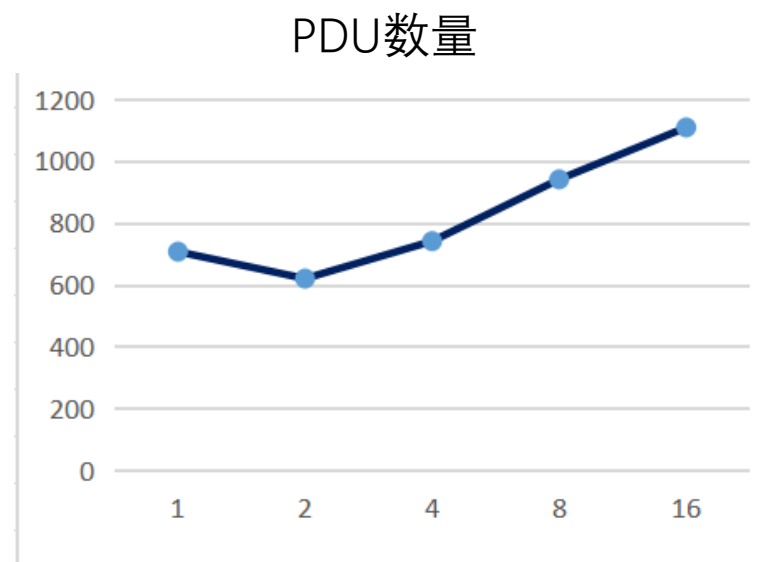
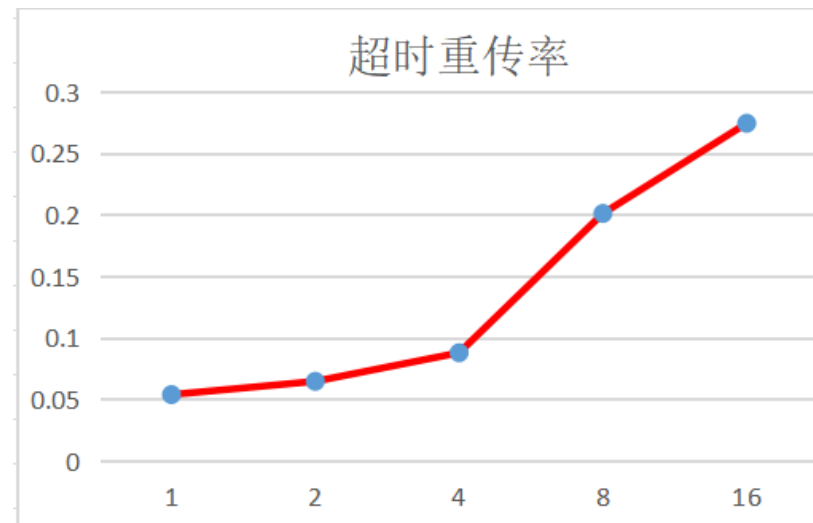
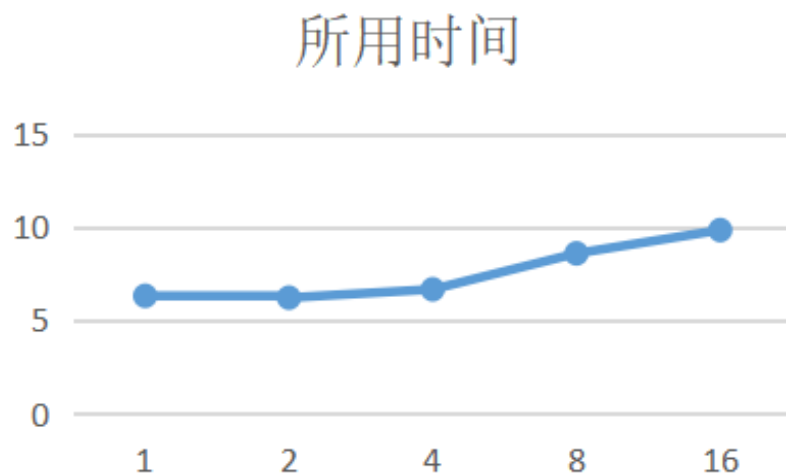
```
225, receive: pdu_exp=6, pdu_rcv=0, status=NoErr
226, receive: pdu_exp=6, pdu_rcv=1, status=NoErr
```

```
215, receive: pdu_exp=2, pdu_rcv=2, status=DataErr
```

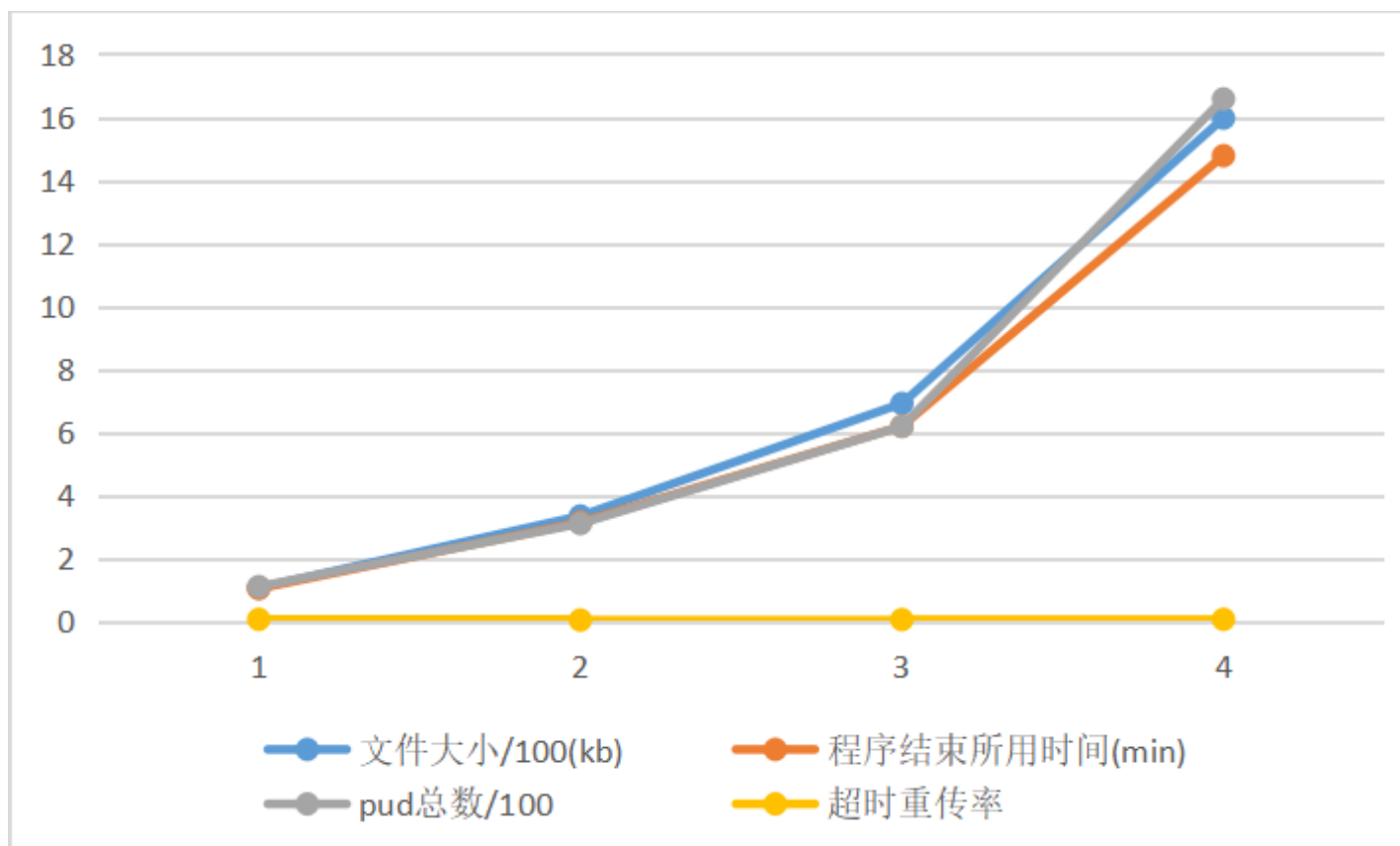


# 结果分析 (Go-Back-N)

- 窗口大小

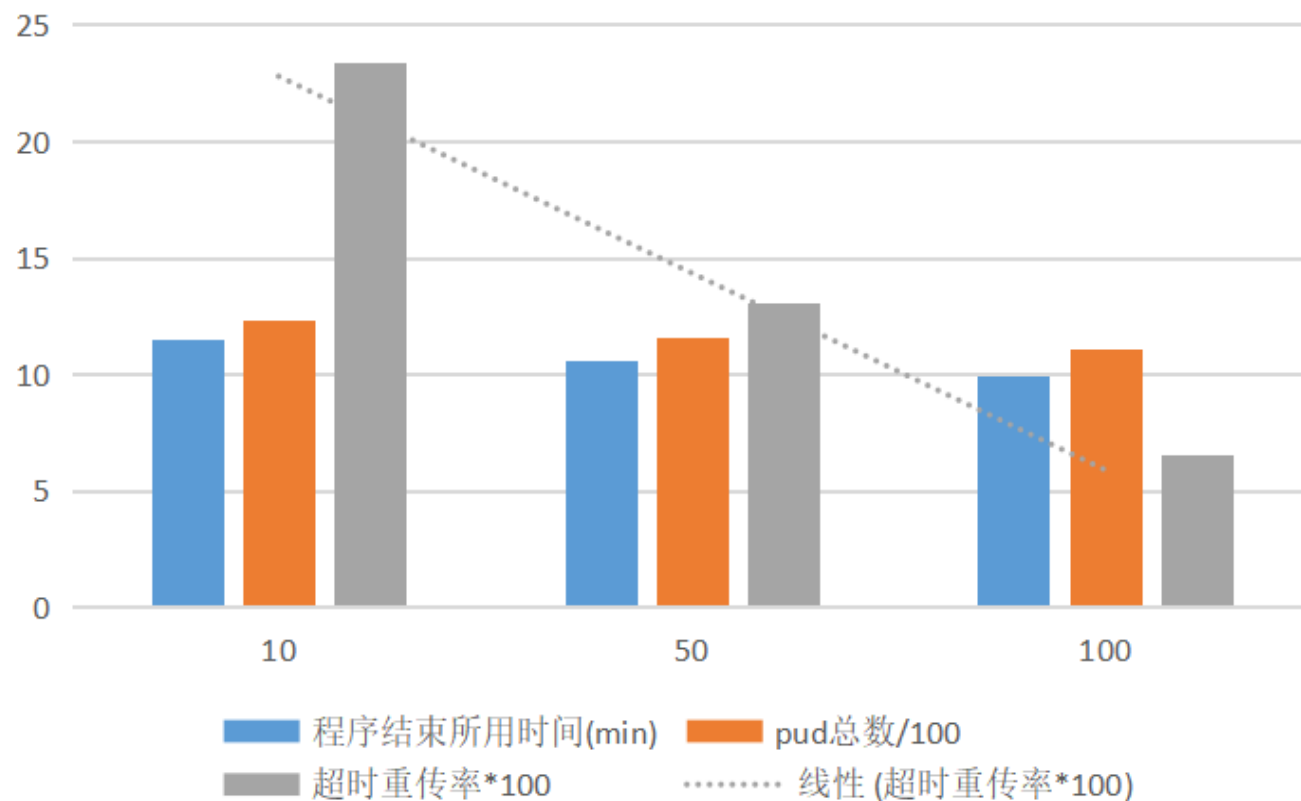


# 结果分析 (Go-Back-N)



# 结果分析 (Go-Back-N)

- 错误率



# Implementing a Distance Vector Routing Protocol

# 项目简介

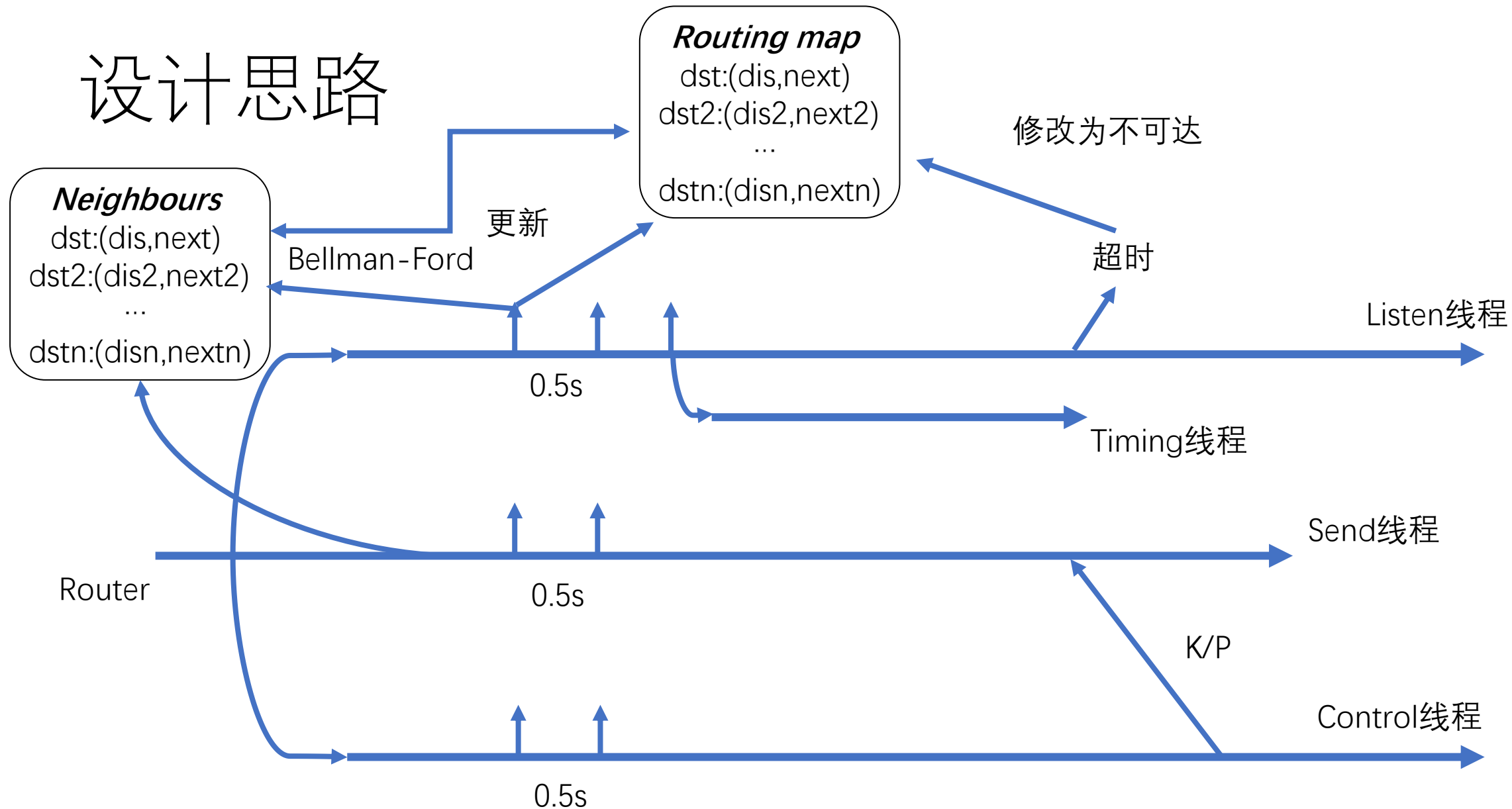
- 使用三种语言实现基于DV算法的路由选择协议
- 可以根据网络状态定期更新路由表
- 可修改网络状态
  - 节点的启动和暂停
  - 拓扑结构链路权重的改变

# 设计思路

每个路由器使用多个线程实现功能：

- 监听：接收距离向量的更新
  - 阻塞接收 0.5秒一次
  - 监听线程内部另开启一个计时线程
- 发送：发送距离向量的更新
  - 时隙，0.5秒发送一次，保证同步
- 控制：读取控制台输入
- 节点暂停：不发送路由表
- 节点关闭：退出循环，结束线程
- 结点重启：读取配置文件，开始发送路由表
- 链路改变：更新对应的距离向量即可

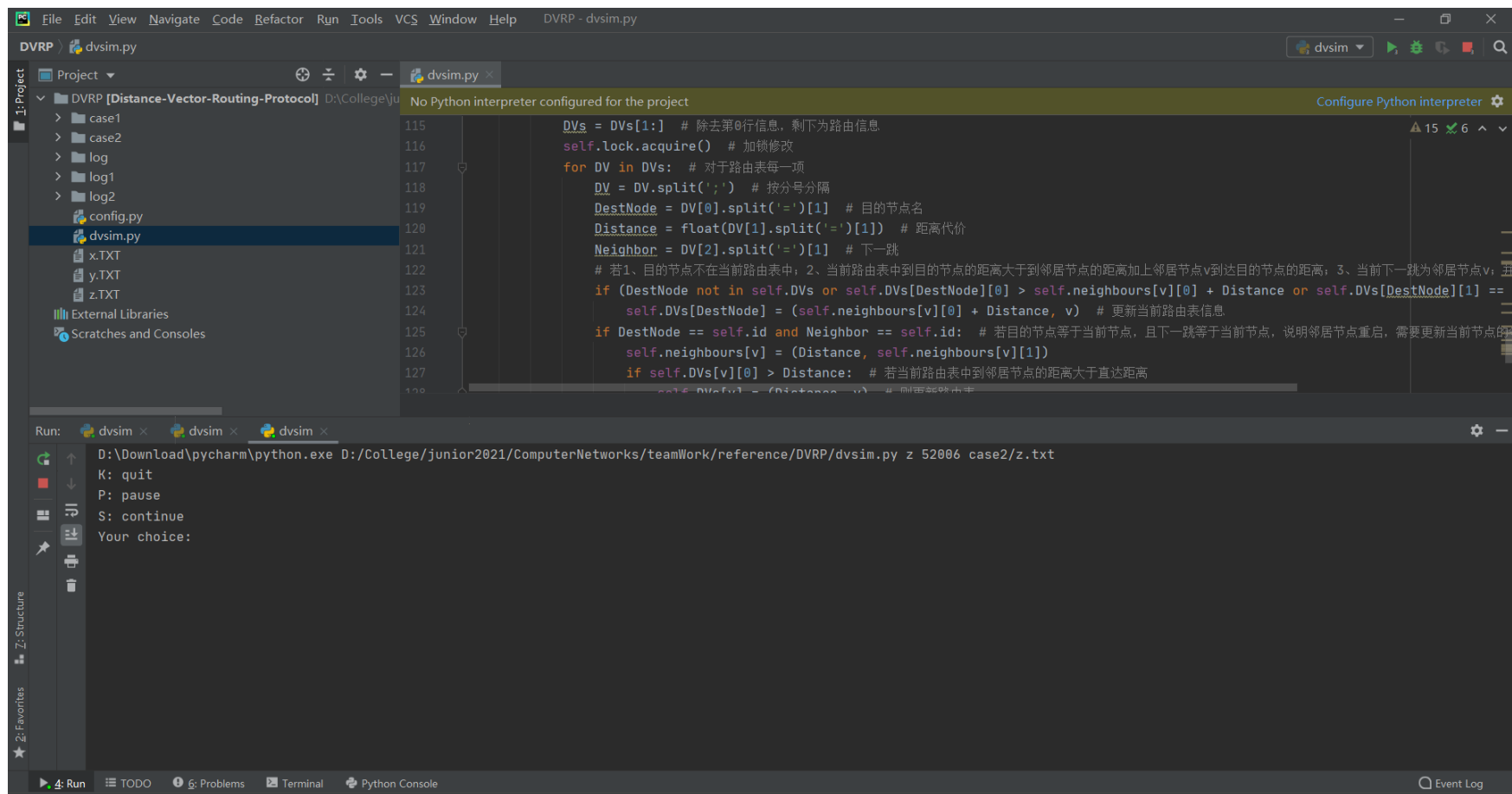
# 设计思路



# 程序演示



# 结果分析(好消息传的快, 坏消息传的慢)



The screenshot displays the PyCharm IDE interface for a project named 'DVRP [Distance-Vector-Routing-Protocol]'. The left sidebar shows the project structure with files like 'config.py', 'dvsim.py', and 'x.TXT', 'y.TXT', 'z.TXT'. The main editor window shows the code for 'dvsim.py', which implements a Distance-Vector Routing Protocol. The code includes comments in Chinese and logic for handling distance vectors (DVs) and neighbors. The bottom panel shows the 'Run' output, indicating the script was executed successfully with the command: `D:\Download\pycharm\python.exe D:/College/junior2021/ComputerNetworks/teamWork/reference/DVRP/dvsim.py z 52006 case2/z.txt`. The output shows the script running and waiting for user input (K: quit, P: pause, S: continue, Your choice:).

```
115 DVs = DVs[1:] # 除去第0行信息, 剩下为路由信息
116 self.lock.acquire() # 加锁修改
117 for DV in DVs: # 对于路由表每一项
118     DV = DV.split(';') # 按分号分隔
119     DestNode = DV[0].split('=')[1] # 目的节点名
120     Distance = float(DV[1].split('=')[1]) # 距离代价
121     Neighbor = DV[2].split('=')[1] # 下一跳
122     # 若1、目的节点不在当前路由表中; 2、当前路由表中到目的节点的距离大于到邻居节点v到达目的节点的距离; 3、当前下一跳为邻居节点v, 且
123     if (DestNode not in self.DVs or self.DVs[DestNode][0] > self.neighbours[v][0] + Distance or self.DVs[DestNode][1] == Neighbor):
124         self.DVs[DestNode] = (self.neighbours[v][0] + Distance, v) # 更新当前路由表信息
125     if DestNode == self.id and Neighbor == self.id: # 若目的节点等于当前节点, 且下一跳等于当前节点, 说明邻居节点重启, 需要更新当前节点的路由表
126         self.neighbours[v] = (Distance, self.neighbours[v][1])
127     if self.DVs[v][0] > Distance: # 若当前路由表中到邻居节点的距离大于直达距离
128         self.DVs[v] = (Distance, v) # 侧直距离由主
```

Run: dvsim x dvsim x dvsim x  
D:\Download\pycharm\python.exe D:/College/junior2021/ComputerNetworks/teamWork/reference/DVRP/dvsim.py z 52006 case2/z.txt  
K: quit  
P: pause  
S: continue  
Your choice:

# 结果分析(好消息传的快, 坏消息传的慢)

- X-log

Received.Source Node=y; Sequence Number=14  
DestNode=x; Distance=18.0; Neighbor=z  
DestNode=z; Distance=1.0; Neighbor=z

y在反复更新

Received.Source Node=z; Sequence Number=15  
DestNode=x; Distance=19.0; Neighbor=y  
DestNode=y; Distance=1.0; Neighbor=y

z在反复更新

Sent.Source Node=x; Sequence Number=2  
DestNode=y; Distance=51.0; Neighbor=z  
DestNode=z; Distance=50.0; Neighbor=z

Received.Source Node=y; Sequence Number=15  
DestNode=x; Distance=20.0; Neighbor=z  
DestNode=z; Distance=1.0; Neighbor=z

y在反复更新

Received.Source Node=z; Sequence Number=16  
DestNode=x; Distance=21.0; Neighbor=y  
DestNode=y; Distance=1.0; Neighbor=y

z在反复更新