



# 区块链技术与应用

v0.11.5

## 实验三：P2P 网络（下）

丁烨，网络空间安全学院 副教授

[dingye@dgut.edu.cn](mailto:dingye@dgut.edu.cn)



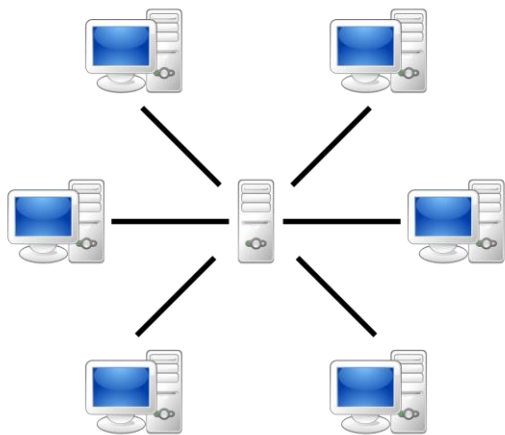
# P2P 网络概述

## P2P 网络技术简介

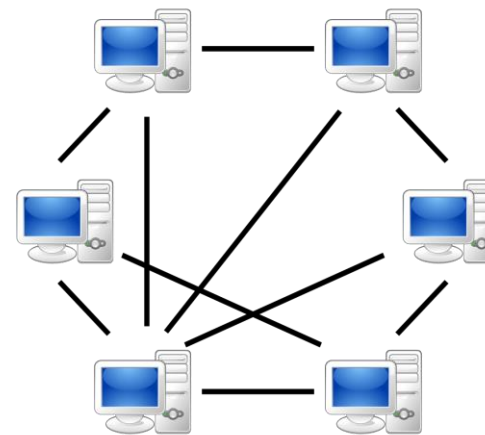
- ❖ 对等式网络（Peer-to-peer, P2P）
- ❖ 又称点对点技术
- ❖ 无中心服务器、依靠用户群（Peers）交换信息的互联网体系
- ❖ P2P 网络可以降低因中心节点出错而导致资料丢失的风险
- ❖ 与有中心服务器的中央网络系统不同，对等网络的每个用户端既是一个节点，也有服务器的功能，任何一个节点无法直接找到其他节点，必须依靠其户群进行信息交流

# P2P 网络概述

## P2P 网络技术简介



中心式网络



P2P 网络

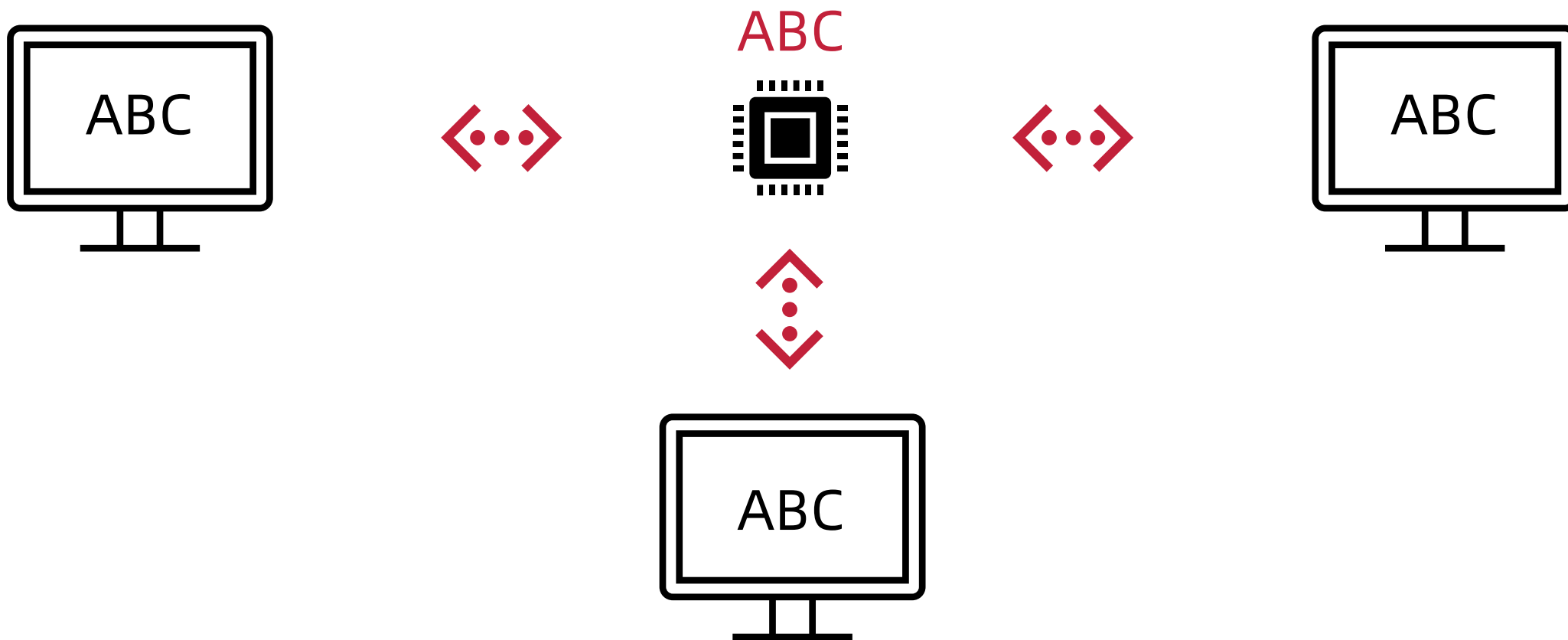
# P2P 网络概述

## P2P 网络的必要性

- ❖ 无中心的 P2P 网络可以防止中心服务器数据污染
- ❖ 从而达到区块链的**安全性**目的
- ❖ 真实的区块链系统中，P2P 网络协议也算“共识”的一种

# 实现简单的 P2P 区块链网络

## P2P 发现协议



# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
import argparse
import json
import logging
import threading
import time
import traceback
```



```
import websocket
from wrenchbox.logging import setup_log
```

```
DEFAULTS = {'max_connection': 3}
```

# 实现简单的 P2P 区块链网络


## 实现简易的 Tracker

### ❖ tracker.py

```
import argparse
import json
import logging
import threading
import time
import traceback

import websocket
from wrenchbox.logging import setup_log

DEFAULTS = {'max_connection': 3}
```



# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
import argparse
import json
import logging
import threading
import time
import traceback

import websocket
from wrenchbox.logging import setup_log

DEFAULTS = {'max_connection': 3}
```





# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
if __name__ == '__main__':  
    parser = argparse.ArgumentParser()  
    parser.add_argument('--debug',  
                        action='store_true', default=False, help='show debug information')  
    parser.add_argument('-t', '--sleep',  
                        type=int, default=30, help='refresh rate in seconds, default: 30')  
    parser.add_argument('seed',  
                        type=str, help='seed announce server, e.g.: ws://localhost:9000/ws')  
    args, _ = parser.parse_known_args()  
    setup_log(level=logging.DEBUG if args.debug else logging.INFO)  
    Tracker().run(args.seed, args.sleep)
```



# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
class Tracker:  
    def __init__(self):  
        self.spawning = []  
        self.peers = []
```

...

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
class Tracker:
```

```
...
```

```
def run(self, seed: str, sleep: int = 30):
```

```
    self.spawn(seed)
```

```
    while True:
```

```
        time.sleep(sleep)
```

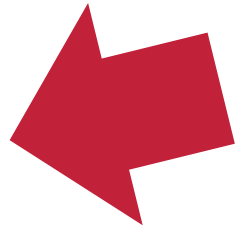
```
        self.query()
```

```
        self.announce()
```

```
        if not len(self.peers):
```

```
            logging.critical('All peers are gone, updater will terminate.')
```

```
            break
```



# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
def spawn(self, url):
    if url not in self.spawning:
        self.spawning.append(url)
        logging.info('Spawning new peer: %s', url)
        ws = websocket.WebSocketApp(
            ...
        )
        peer = threading.Thread(target=ws.run_forever)
        peer.daemon = True
        peer.start()
        self.spawning.remove(url)
```

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
ws = websocket.WebSocketApp(  
    url=url,  
    on_open=self.on_open,  
    on_message=self.on_message,  
    on_close=self.on_close  
)
```

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
def on_open(self, ws):  
    logging.info("New peer connected: %s", ws.url)  
    self.peers.append(ws)
```

```
def on_close(self, ws):  
    logging.info("Peer disconnected: %s", ws.url)  
    if ws in self.peers:  
        self.peers.remove(ws)
```

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
def on_message(self, ws, message):
    if len(self.peers) >= DEFAULTS['max_connection']:
        return
    try:
        message = json.loads(message)
    except json.JSONDecodeError:
        pass
    if 'response' in message and 'peers' in message['response']:
        for peer in message['response']['peers']:
            if peer not in [i.url for i in self.peers]:
                self.spawn(peer)
```

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

### ❖ tracker.py

```
def query(self):
    for peer in self.peers:
        try:
            peer.send(json.dumps({
                'op': 'peers'
            }))
        except:
            logging.error('Cannot request peers from: %s', peer.url)
            traceback.print_exc()
```



# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

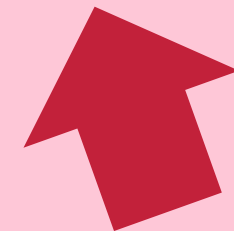
### ❖ tracker.py

```
def announce(self):
    for peer in self.peers:
        try:
            peer.send(json.dumps({
                'op': 'register',
                'args': {
                    'addr': [i.url for i in self.peers]
                }
            }))
        except:
            pass
```

# 实现简单的 P2P 区块链网络

## 实现简易的 Tracker

```
~/Documents/Workspace/foxchain(master*) » python3 -u -m foxchain.app.tracker -t 5 ws://localhost:9001/ws  
[2021-04-09 18:28:01,936] Spawning new peer: ws://localhost:9001/ws  
[2021-04-09 18:28:01,943] New peer connected: ws://localhost:9001/ws  
[2021-04-09 18:28:26,957] Spawning new peer: ws://localhost:9002/ws  
[2021-04-09 18:28:26,961] New peer connected: ws://localhost:9002/ws
```



Everything is ok, ready to go!

ws://localhost:9001/ws

Disconnect

Send

```
1 {  
2   "op": "register",  
3   "args": {  
4     "addr": "ws://localhost:9002/ws"  
5   }  
6 }
```

```
1 {  
2   "status": 202,  
3   "error": "Accepted"  
4 }
```

Everything is ok, ready to go!


ws://localhost:9001/ws

Disconnect

Send

```
1 {  
2   "op": "peers"  
3 }
```

```
1 {  
2   "status": 200,  
3   "error": "OK",  
4   "response": {  
5     "peers": ["ws://localhost:9001/ws", "ws://localhost:9002/ws"]  
6   }  
7 }
```



Everything is ok, ready to go!

ws://localhost:9002/ws

Disconnect

Send

```
1 {  
2   "op": "peers"  
3 }
```

```
1 {  
2   "status": 200,  
3   "error": "OK",  
4   "response": {  
5     "peers": ["ws://localhost:9001/ws", "ws://localhost:9002/ws"]  
6   }  
7 }
```

# 作业

- ❖ 参考实验教程搭建一个 P2P 服务
- ❖ 在三个以上的节点上测试 P2P 同步效果
- ❖ 提供完整的测试代码及截图

# 作业

- ❖ 在作业系统中下载并完成本实验课对应实验报告
- ❖ <https://hw.css.dgut.edu.cn/>
- ❖ 注意：所有标识为 \* 的地方都需要填写
- ❖ 请务必在截止时间之前提交实验报告

课程名称：区块链技术与应用

学期：2021 年春季

实验名称	分布式账本			实验序号	1
姓名	***	学号	***	班级	***
实验地点	***	实验日期	***	指导老师	丁焯
教师评语	-			实验成绩	-
				百分制	100
同组同学					

## 四、实验作业及分析

### 4.1 实验过程

1) \*\*\* 请将详细实验过程填写在此处 \*\*\*

### 4.2 实验结果

\*\*\* 请将实验结果截图填写在此处 \*\*\*

## 五、实验总结

\*\*\* 请撰写一段 200 字左右的实验总结 \*\*\*

