



区块链技术与应用

v0.10.29

实验二：P2P 网络（上）

丁烨，网络空间安全学院 副教授

dingye@dgut.edu.cn



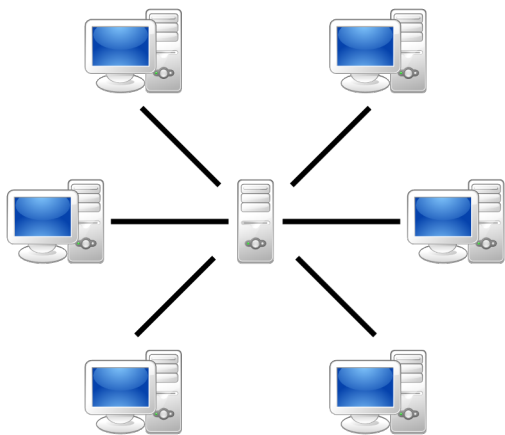
P2P 网络概述

P2P 网络技术简介

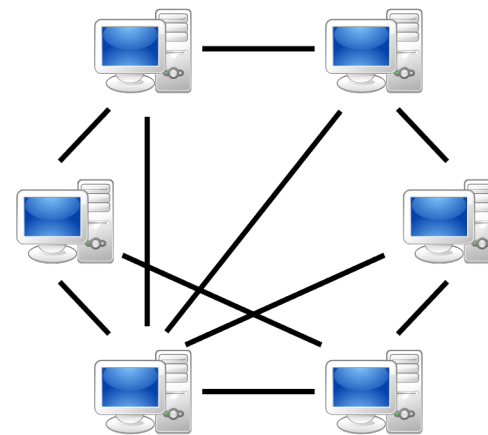
- ❖ 对等式网络（Peer-to-peer, P2P）
- ❖ 又称点对点技术
- ❖ 无中心服务器、依靠用户群（Peers）交换信息的互联网体系
- ❖ P2P 网络可以降低因中心节点出错而导致资料丢失的风险
- ❖ 与有中心服务器的中央网络系统不同，对等网络的每个用户端既是一个节点，也有服务器的功能，任何一个节点无法直接找到其他节点，必须依靠其户群进行信息交流

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

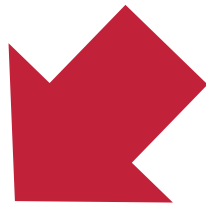
P2P 网络的必要性

- ❖ 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

简易 P2P 节点

❖ app.py



```
import argparse
import json
import logging
import pickle
```

```
from redislite import Redis
from tornado import websocket, web, ioloop
from wrenchbox.logging import setup_log
```


```
DEFAULTS = {'port': 9000}
clients = []
db = None
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if __name__ == '__main__':  
    parser = argparse.ArgumentParser()  
    parser.add_argument(  
        '--debug',  
        action='store_true',  
        default=False, help='show debug information'  
    )
```



实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py


```
if __name__ == '__main__':  
    ...  
    parser.add_argument(  
        '-p', '--port',  
        type=int,  
        default=DEFAULTS['port'],  
        help='listening port, default: {}'.format(DEFAULTS['port'])  
    )
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if __name__ == '__main__':  
    ...  
    parser.add_argument(  
        '-r', '--redis',  
        type=str,  
        default='redis.db',  
        help='redis database file, default: redis.db'  
    )  
    args, _ = parser.parse_known_args()  
    print(args)
```



实现简单的 P2P 区块链网络

简易 P2P 节点

```
~/Documents/Workspace/foxchain(master) » python3 -u -m foxchain.app.app  
Namespace(debug=False, port=9000, redis='redis.db')
```



实现简单的 P2P 区块链网络

简易 P2P 节点

```
-----  
~/Documents/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app -h  
usage: app.py [-h] [--debug] [-p PORT] [-r REDIS]
```

optional arguments:

-h, --help	show this help message and exit
--debug	show debug information
-p PORT, --port PORT	listening port, default: 9000
-r REDIS, --redis REDIS	redis database file, default: redis.db


```
-----
```



实现简单的 P2P 区块链网络

简易 P2P 节点

```
~/Documents/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app -p 3000 -r test.db  
Namespace(debug=False, port=3000, redis='test.db')
```




实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
import argparse  
import json  
import logging  
import pickle
```



```
from redislite import Redis  
from tornado import websocket, web, ioloop  
from wrenchbox.logging import setup_log
```

```
DEFAULTS = {'port': 9000}  
clients = []  
db = None
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
import argparse
import json
import logging
import pickle
```

```
from redislite import Redis
from tornado import websocket, web, ioloop
from wrenchbox.logging import setup_log
```



```
DEFAULTS = {'port': 9000}
clients = []
db = None
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if __name__ == '__main__':  
    ...  
    setup_log(level=logging.DEBUG if args.debug else logging.INFO)  
    db = Redis(args.redis)  
    if b'peers' not in db.keys():  
        db.set('peers', pickle.dumps(set([])))
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
import argparse
import json
import logging
import pickle
```

```
from redis import Redis
from tornado import websocket, web, ioloop
from wrenchbox.logging import setup_log
```


```
DEFAULTS = {'port': 9000}
clients = []
db = None
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if __name__ == '__main__':  
    ...  
    web.Application([  
        (r'/', IndexHandler),  
        (r'/ws', SocketHandler)  
    ]).listen(args.port)  
    logging.info('Tornado is listening on port: %d', args.port)  
    ioloop.IOLoop.instance().start()
```



实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
class IndexHandler(web.RequestHandler):  
    def get(self):  
        self.render("index.html")
```

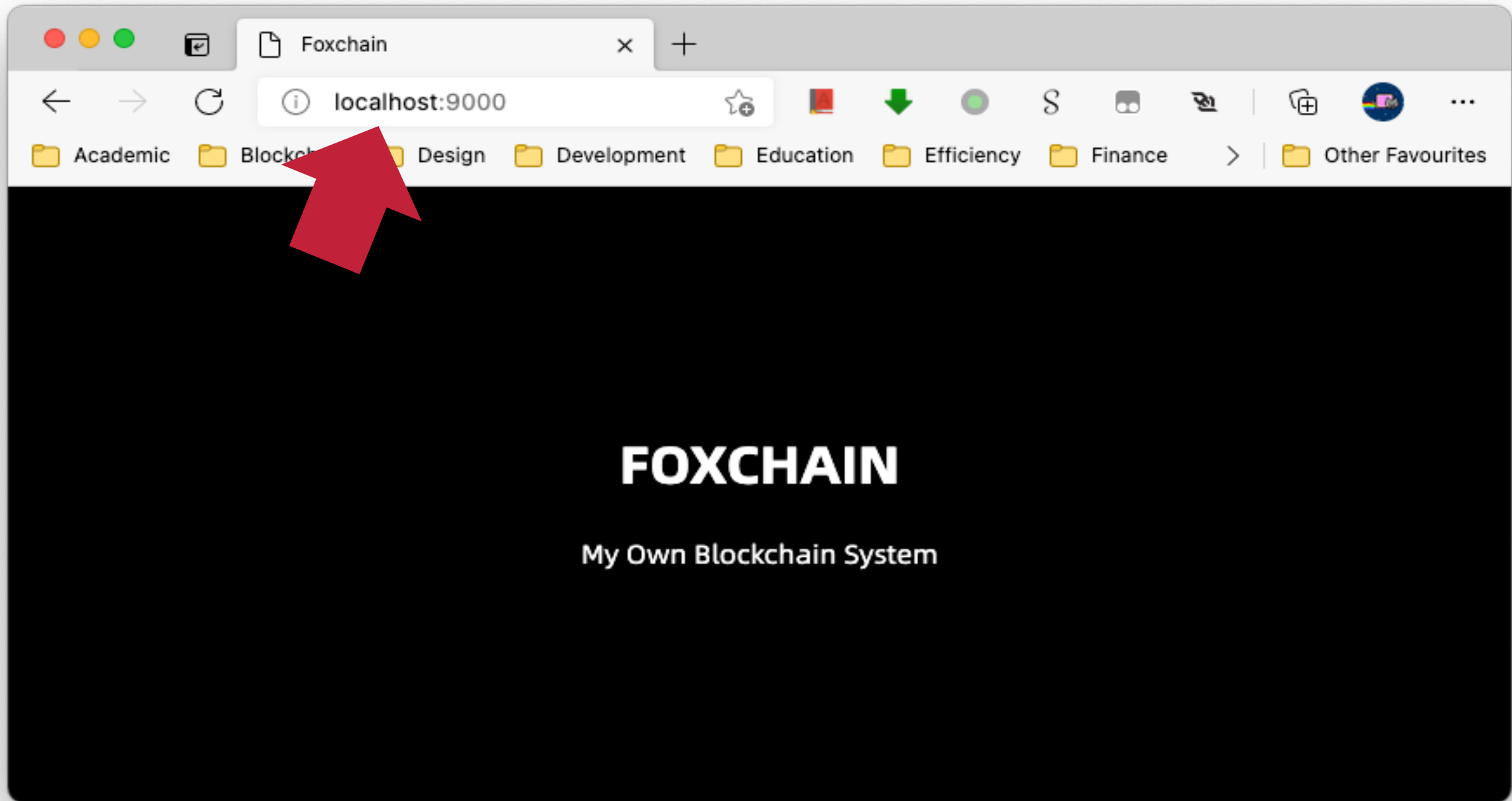
❖ index.html

```
<div class="container">  
    <h1 class="pixel">FOXCHAIN</h1>  
    <p>My Own Blockchain System</p>  
</div>
```

实现简单的 P2P 区块链网络

简易 P2P 节点

```
~/Documents/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app  
[2021-04-01 17:10:26,386] Tornado is listening on port: 9000
```



实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
class SocketHandler(websocket.WebSocketHandler):  
    def check_origin(self, origin):  
        return True  
  
    def open(self):  
        logging.info('Client connected: %s', self.request.remote_ip)  
        if self not in clients:  
            clients.append(self)  
  
    def on_close(self):  
        if self in clients:  
            clients.remove(self)
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
def on_message(self, message):
    try:
        message = json.loads(message)
    except json.JSONDecodeError:
        logging.warning('Cannot parse request message: %s', message)
        self.write_message(json.dumps({
            'status': 500,
            'error': 'Cannot parse request message.',
            'response': None
        }))
    else:
        ...
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if message is not None:
    if 'op' in message:
        if message['op'] == 'register':
            if 'args' in message and 'addr' in message['args']:
                ...
            else:
                ...
        elif message['op'] == 'peers':
            ...
        else: ...
    else: ...
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if 'args' in message and 'addr' in message['args']:
    peers = pickle.loads(db.get('peers'))
    if not isinstance(message['args']['addr'], list):
        message['args']['addr'] = [str(message['args']['addr'])]
    for addr in message['args']['addr']:
        if addr.startswith('ws://') or addr.startswith('wss://'):
            peers.add(addr)
    db.set('peers', pickle.dumps(peers))
    self.write_message(json.dumps({
        'status': 202,
        'error': 'Accepted'
    })))
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
if 'args' in message and 'addr' in message['args']:
    ...
else:
    self.write_message(json.dumps({
        'status': 500,
        'error': 'Operation "register" requires the following "args": "addr"',
        'response': None
    })))
```


实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
elif message['op'] == 'peers':
    self.write_message(json.dumps({
        'status': 200,
        'error': 'OK',
        'response': {'peers': list(pickle.loads(db.get('peers')))}
    }))
else:
    self.write_message(json.dumps({
        'status': 404,
        'error': 'Operation "{}" is not supported.'.format(message['op']),
        'response': None
    })))
```

实现简单的 P2P 区块链网络

简易 P2P 节点

❖ app.py

```
else:  
    logging.warning('Message body is not supported: %s', message)  
    self.write_message(json.dumps({  
        'status': 500,  
        'error': 'Message body is not supported.',  
        'response': None  
    })))
```

实现简单的 P2P 区块链网络

简易 P2P 节点

```
~/Documents/Workspace/foxchain(master*) » python3 -u -m foxchain.app.app  
[2021-04-01 17:10:26,386] Tornado is listening on port: 9000
```

实现简单的 P2P 区块链网络

测试 P2P 节点

- ❖ WebSocket 客户端
- ❖ <http://www.websocket-test.com/>
- ❖ <https://chrome.google.com/webstore/detail/smart-websocket-client/omalebghpgejjiaoknljcfmglgbpocdp>

HISTORY

Clear

Request #72

Request #71

Request #70

Request #69

Request #68

Request #67

Request #66

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

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

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

HISTORY

Clear

Request #73

Request #72

Request #71

Request #70

Request #69

Request #68

Request #67

Request #66

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

```
1 {  
2   "op": "register",  
3   "args": {  
4     "addr": "ws://localhost:9000/ws"  
5   }  
6 }
```

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

HISTORY

Clear

Request #74

Request #73

Request #72

Request #71

Request #70

Request #69

Request #68

Request #67

Request #66

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

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

```
1 {  
2   "status": 200,  
3   "error": "OK",  
4   "response": {  
5     "peers": ["ws://localhost:9000/ws"]  
6   }  
7 }
```

作业

- ❖ 参考实验教程搭建一个 P2P 服务
- ❖ 为 P2P 服务添加一个 OP 功能，使其可以返回节点的时间
- ❖ 提供完整的测试代码及截图

HISTORY

Clear

Request #76

Request #75

Request #74

Request #73

Request #72

Request #71

Request #70

Everything is ok, ready to go!

ws://localhost:9000/ws

Disconnect

Send

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



```
1 {  
2   "status": 200,  
3   "error": "OK",  
4   "response": {  
5     "time": 1617786770.954288,  
6     "time_zone": "CST"  
7   }  
8 }
```



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
- ❖ 截止日期：2021-11-11 23:59

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 ***

