P2P Systems and Blockchains
Spring 2018,
instructor: Laura Ricci
laura.ricci@unipi.it

Lesson 1: introduction
21/2/2018
Prerequisites:
- Computer Networks
- Algorithm design
- Basic concepts of applied cryptography. but we will review these in the course

Instructor: Laura Ricci
Credits: 6
Moodle Link: https://elearning.di.unipi.it/course/view.php?id=118
Timetable:
14:00-16:00 Room Fib N1
14:00-16:00 Room Fib L1

Several guest lectures:
- Paolo Mori: access control system on a blockchain
- Daniele Mazzei: blockchain for IoT
- Pietro Marchionni: agenzia per l'Italia Digitale (Agid)
EXAM RULES

- Mid term + Final term (not compulsory)
  - if the student pass both mid and final term the oral proof, he/she is exempted from the oral examination
  - tentative ideas:
    - mid term: reading a paper and writing a brief relation
    - final term: programming exercise useful for the final project

- Exam: final project
  - project proposals from the students are accepted
  - the project may be the first part of the final thesis
  - oral examination: only for those who have not passed the midterm and/or final term
COURSE STRUCTURE (TENTATIVE)

- Main focus on:
  - design of P2P systems
  - cryptocurrencies (Bitcoin)
  - block-chains: Ethereum

- To define P2P systems, we will exploit
  - distributed algorithms
  - probabilistic data structures
  - formal methods

- attention on practical experience: not a laboratory course, but a final project

- If room: P2P are complex networks, millions of nodes and connections
  - complex network modelling
  - big graph analysis
PRELIMINARY PROGRAM

• **Peer to Peer Topologies**

  • unstructured overlays: Flooding, Random Walks
  
  • structured overlays: Distributed Hash Tables (DHT)
    • The Chord DHT: formal analysis of Chord routing
    • Kademlia
    • applications of DHT
      • Bittorrent: a Content Distribution Network exploiting KAD (based on Kademlia)

• **Information Diffusion**

  • Epidemic protocols
Cryptocurrencies and blockchains

- review of basic cryptographic tools
  - digital signatures
  - cryptographic hash
  - Merkle trees
  - Patricia trees
  - Bloom filters
- consensus
  - Byzantine agreement
  - alternative notions
- the Bitcoin blockchain
  - transactions, block, the blockchain
  - Nakamoto consensus: mining mechanism
  - the Bitcoin P2P Network
  - the Bitcoin ecosystem
  - scalability issues
  - pseudoanonymity: traceability and mixing
PRELIMINARY PROGRAM

• Bitcoin extensions/alternatives
  • altcoins, sidechains,

• Ethereum:
  • contracts, transactions, gas
  • smart contracts programming
    • Solidity

• Applications of blockchains
  • Blockchain 1.0: cryptocurrencies
  • Blockchain 2.0: financial instruments built on cryptocurrencies
  • Blockchain 3.0: applications beyond cryptocurrencies (DNS, lotteries, voting, IoT...)
Complex Network Models and tools

- Random Graphs and Small Worlds
- Small world navigability: Watts Strogatz and Kleinberg Models
- Application to peer to peer networks
  - Chord
  - Darknet Freenet, Symphony

- P2P audio/video Streaming
  - streams diffusion overlays: tree, multi-tree, mesh
  - Skype

- Distributed Online Social Networks

- System issues in P2P System: NAT traversal
DIDACTIC MATERIAL

Mandatory

- lesson slides (download this year slides)
- tutorial and material published on the course page

Reference books:

- Sasu Takoma, *Overlay Networks*, Toward Information Networking, 2010


- Maarten Van Steen, *Graph Theory and Complex Networks*, 2010
Further references:


TOOLS FOR THE PROJECT

• Peersim
  • a simulator oriented to P2P simulations
  • highly scalable (until 10000 peers for the implementation of simple protocols)
  • http://peersim.sourceforge.net/
  • a lesson to introduce this tool, full support from my team

• Ethereum
  • Programming smart contracts
  • Test net
  • Remix (http://remix.readthedocs.io/en/latest/)
PROJECT PROPOSALS

- Simulation of
  - unstructured P2P networks
  - distributed Hash Tables (DHT)
  - self-organizing systems

with the goal of analysing P2P topologies

- Writing smart contract in Ethereum for
  - IoT
  - Identity services
  - Voting

exploiting the Ethereum test net
THE CLIENT SERVER PARADIGM

- runs on end-hosts
- on/off behavior
- service consumer
- issue requests
- do not communicate directly among them
- need to know the server IP address

- runs on end-hosts
- always on
- service provider
- receive requests
- satisfies all client requests
- need a fixed IP (or DNS name)
THE PEER TO PEER PARADIGM

Peer

Runs on end-hosts

- on/off behavior, handle **churn**
- need to join
- need to discover other peers
- service providers and consumers
- communicate directly among them
- need to define communication rules
  - prevent free riding
  - incentivate participation ... and
  - reciprocation
Notice that:

- servers are typically needed for bootstrap
- but servers aren’t needed for resource sharing
THE PEER TO PEER PARADIGM

- Runs on end-hosts
- On/off behavior, handle churn
- Need to join
- Need to discover other peers
- Service providers and consumers
- Communicate directly among them
- Need to define communication rules
- Prevent free riding
- Incentivate participation and reciprocation

Peer

Servers used for bootstrap
Definition 1:
A peer to peer system is a set of autonomous entities (peers) able to auto-organize and sharing a set of distributed resources in a computer network. The system exploits such resources to give a service in a complete or partial decentralized way.

Shared Resources:
- Read Only Information (Files)
- Read/Write storage space (Distributed File System)
- Computing power
- Bandwidth
Definition 2:

A P2P system is a distributed system defined by a set on nodes interconnected able to auto-organize and to build different topologies with the goal of sharing resources like CPU cycles, memory, bandwidth. The system is able to adapt to a continuous churn of the nodes maintaining connectivity and reasonable performances without a centralized entity (like a server).
RESOURCE SHARING

- P2P: is relative to **give** and **receive** from a community. Each peer shares a set of resources and obtains, in return, a set of resources
  - the most common scenario: share musics (audio files) and obtain music in return (Napster, Gnutella, …)
  - a peer behaves both like a client and like a server (symmetric functionality = Servent)

- but a peer can decide of **offering for free** a resource, for instance to participate to a project
  - searching extra-terrestrial life
  - cancer therapies research
  - contributing to the maintenance of a **distributed ledger**

- the shared resources are at “the border” of Internet, they are directly shared by the peers, there are no “special purpose nodes” for their management.
RESOURCE SHARING

- the peers' connection is **transient**: the connections and disconnections to the network are very frequent

- the resources offered by the peers are **dynamically** added and removed

- each peer is paired with a different IP address for each connection to the system

  ⇒

  a resource cannot be located by a static IP

  ⇒

  new addressing mechanisms have to be defined, at the application level, not at the IP level
P2P: FILE SHARING AND BEYOND...

P2P file sharing
• file sharing: light weight/ best effort
• persistence and security are not the main goal
• anonymity is important
• Examples:
  • Napster
  • Gnutella, KaZaa
  • eMule
  • BitTorrent
• P2P Media Streaming
  • Skype (first versions)
  • IPTV: video streaming applications
• Cryptocurrencies and blockchains
A P2P file sharing application

- A user U has a P2P client on its notebook
- The connection of the to Internet is intermittent: each time the user obtains a new IP address for each new connection
- The user stores the shared files in a directory and pairs each file with a set of keys able to identify it (for a song: title, author, publication date,...)
- U is interested in finding a song and sends a query to the system
- The client finds and shows to the user information about the other peers which own the required song
- U chooses a peer P among these (according some criteria, we will see in the following)
- The file is copied from the notebook of P to that of U
- While U is performing the download, other users can upload the parts of the file already downloaded by U and put in the shared directory
FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

• File Sharing: begins with the rapid success of Napster, at the end of nineties, about 10 years later than Wide Web

• First generation: Napster
  • introduces a set of servers where the users register the *descriptors* of the files they are going to share
  • only the content transmission (download/upload) exploits a P2P protocol, content search is centralized
  • the presence of a centralized directory has been the 'Achille's heel' of this application
  • it was convicted because it did not respect the law on the copyright
    • it could have detected the content exchanged between the users through the analysis of the centralized directory
FILE SHARING: THE FIRST P2P 'KILLER APPLICATION'

- File sharing: second generation
  - no centralization point
  - both file search and content transfer are completely distributed
  - Gnutella, FastTrack/Kazaa, BitTorrent

- Side effects of the diffusion of file sharing applications: change of the way users music is accessed by the
  - from CD to online music
  - iTunes
The P2P client behaves like a servlet and allows:

- the user to define a directory, in its file system, where the file it is going to share are stored. Each other peer of the P2P network may download files from this directory
  - a peer behaves like a web server

- the user to download files from the shared directory
  - a peer behaves like a client

- the user to find the content it is interested in, through queries submitted to the system
  - this functionality is similar to that of Google
MEDIA STREAMING

• Media content distributed on net and played out while downloading
  • No need for a complete download to begin the playback

• Live media streaming
  • Media produced on the go (e.g. live concert, sport match)
  • approximate synchronicity among stream production, download and playback
    • Application contraints, e.g., conferencing requires lower delays than Internet TV

• On demand streaming
  • Content stored in a server and sent upon request
  • Features allowed such as pause, fast forward, fast rewind
  • Possible to download the stream at a rate higher than production/playback rate
P2P MEDIA STREAMING

- P2P file sharing has no issues like QoS:
  - No need to playback the media in real time
  - Downloading takes long time, many users do it overnight
- Media streaming extremely expensive
  - 1 hour of video encoded at 300Kbps = 128.7 MB
  - Serving 1000 users would require 125.68 GB
- Media Server cannot serve everybody in swarm
- In P2P Streaming:
  - Peers form an overlay of nodes at the application level
  - Nodes in the overlay connected by direct paths (virtual or logical links), in reality, connected by many physical links in the underlying network
  - Nodes offer their uplink bandwidth while downloading and viewing the media content
  - Takes load off the server
  - Scalable
P2P MEDIA STREAMING

- Quality of Service requirements of media streaming application
  - High bandwidth
  - low latency
  - low loss rate

- High “churn” rate – Users join and leave in between
  - Needs robust network topology to overcome churn

- Internet dynamics and congestion in the interior of the network
  - degrades QoS

- Fairness policies extremely difficult to apply
  - High bandwidth users have no incentive to contribute
  - Bitcoin Tit-for-tat policy difficult to be applied
P2P MEDIA STREAMING

- SplitStream
- DONet/Coolsteraming
- CoopNet
- Orchard
- Bullet
- Prime
- Pulsar
- NICE
- Zigzag
- DirectStream
- MeshCast

- mtreeBone
- PULSE
- GnuStream
- SAAR
- ChainSaw
- ChunkySpread
- BulkTree
- ForestCast
- AnySee
- DagStream
- Climber

- CollectCast
- HyMoNet
- GridMedia
- Promise
- Yoid
- Zebra
- Tribler
- CliqueStream
- GradienTv
- Sepidar
The Times 03/Jan/2009 Chancellor on brink of second bailout for banks.

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of

Paper published in October 2008: good birthday, Bitcoin!
WHY STUDY CRYPTOCURRENCIES?

- Money: it’s an important topic, it’s all around us.
- We can learn a lot about computer security from observing real life cryptocurrencies.
- The cypherpunk vision: we can revolutionize our world by building secure protocols.
- New motivation and tools for learning traditional concepts in computer security.
- “Blockchain technology,” a related and more general concept:
  - A new technology for developing secure applications in an untrusted environment.
  - Ethereum.
  - Affects many industries.
THE BITCOIN ECOSYSTEM

- a peer-to-peer version of the classical electronic payment
- payments directly done between the users, no a centralized financial entity which guarantees the electronic payment, lower costs
ETHEREUM

- Crowdfunded ~$20M in ~ a month
- Popularized a grand vision of “generalized” cryptocurrency
- Smart contracts:
  - implement a protocol that uses a block-chain
  - can be used by multiple users who do not trust each other
  - programmable through Turing complete language
    - Solidity
    - Serpent,...
P2P CONTENT DISTRIBUTION

A server publishes new content (example: a new game release, a new release of an operating system, a song...)

- client server model: a single centralized
- server is a bottleneck,
- especially for video/audio streaming applications

• P2P approaches obtain a better balance of the use of the bandwidth by exploiting the less used communication channels
P2P CONTENT DISTRIBUTION

- P2P approaches
  - to obtain a better balance of the use of the communication bandwidth by exploiting the less used communication channels
  - shared resource is the bandwidth

- Peer-to-Peer Content Distribution
  - the initial file request are served by a centralized server
  - further requests are sent to the peers which have already received and replicated the files
A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server.

- Content distribution networks (for instance Bittorrent)
A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server

- Content distribution networks (for instance Bittorrent)
A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server

- Content distribution networks (for instance Bittorrent)
A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server

- Content distribution networks (for instance Bittorrent)
A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server

- Content distribution networks (for instance Bittorrent)
P2P CONTENT DISTRIBUTION

- A combined use of the P2P and of the client-server model enables the optimization of the accesses to the server
- Segmented approach (example: BitTorrent)
A P2P Storage Network is a cluster of computers connected in the network and exploits all the memory shared by the peers for the definition of a distributed storage system.

- Examples: DHT, often store read a read only index
- Real distributed file systems: PAST, Freenet, OceanStore, Wuala

Organization:
- Peer-identifiers association through a hash function
- Each peer offers a part of its storage space or pays an amount of money
- A maximum volume of data are assigned to each peer, according to the contribution of the peer to the storage network.
- Document-identifier assignment computed through a hash function computed on the name of the file or on a part of its content
- The storage and search of files in the network is guided by the identifiers paired with the hash function with peers and files.
Storage Network Definition
P2P STORAGE NETWORKS

Storage Network Definition

- Hash ID 1
- Hello ???
- ID 4
- Hello ???
- ID 25
- Hello ???
- ID 17
- ID 8
- ID 3
- ID 10
Storage Network Definition

P2P STORAGE NETWORKS

Hash

ID 1

ID 25

ID 3

Hello

ID 4

ID 4

ID 17

ID 8

ID 10

ID 3
Storage Network Definition

Hash

ID 1

ID 3

ID 17

ID 8

ID 25

ID 4

ID 4

ID 4

ID 25

ID 25

ID 3

ID 10

ID 3
Document Storage

P2P STORAGE NETWORKS
Document Storage

Hash

ID 1

ID 3

ID 4

ID 10

ID 11

ID 25

ID 17

ID 25

ID 8

ID 17

ID 10

ID 8
Document Storage

P2P STORAGE NETWORKS

Dipartimento di Informatica
Università degli Studi di Pisa

P2P Systems and Blockchains: Introduction
Laura Ricci
P2P STORAGE NETWORKS

Document Storage

Hash

ID 1

ID 3

ID 4

ID 25

ID 10

ID 17

ID 8

ID 1

ID 4

ID 10

ID 8

ID 3

ID 25

ID 17

ID 25

ID 3

ID 8

ID 17

ID 10
Document Storage
Document Storage

P2P STORAGE NETWORKS

Dipartimento di Informatica
Università degli Studi di Pisa

Laura Ricci

P2P Systems and Blockchains: Introduction

53
Document Storage

P2P STORAGE NETWORKS

Dipartimento di Informatica
Università degli Studi di Pisa

P2P Systems and Blockchains: Introduction
Laura Ricci
P2P STORAGE NETWORKS

Document Storage

Hash

ID 1

ID 11

ID 4

ID 25

ID 11

ID 4

ID 17

ID 25

ID 11

ID 3

ID 10

ID 17

ID 8

ID 25

ID 10

ID 17

ID 8

ID 3

ID 4

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10

ID 17

ID 8

ID 3

ID 10
Document Storage
Document Storage
Document Storage
P2P STORAGE NETWORKS

Document Storage

ID 1

ID 3

ID 4

ID 17

ID 25

ID 11

ID 10

ID 8

ID 3

ID 4

ID 10

ID 17

ID 25

ID 11

requestor: 1
P2P STORAGE NETWORKS

- A P2P Storage System exploits all the storage offered by the peers
  - PAST, Freenet, OceanStore,
  - **Freenet**: mechanism for anonymisation
    - Substitution of the sender of a message
    - Encrypting Techniques

- **Distributed Hash Tables techniques**:
  - Each peer offers a part of its storage or pays an amount of money
  - According to its contribute, a maximum volume of data is assigned to each peer
  - **Hash functions** to assign
    - identifiers to peers
    - identifiers to documents computed by exploiting the file name or part of its content
  - The storage and search of the files is guided by the identifiers computed by the hash function.
P2P networks are commonly called overlays (Overlay Network): logic network connecting peers laid over the IP infrastructure, i.e. defined at application level

- the overlay defines a set of logic links between the peers, which does not correspond to physical connections
- each logical link corresponds to:
  - a set of physical links: not all logical links are also physical
  - not all physical links are necessarily used
  - transversal of a set of routers
P2P OVERLAY

- Overlay network: a network of peers built on top of an existing network
  - virtual (logical) network
    - made up of nodes and logical links
    - nodes are application processes/threads
  - built on top of an existing network
    - logical links that can span many physical links
  - to provide a service not available in the underlying network
    - the overlay relies on underlays for basic networking (i.e., routing/forwarding)
    - can offer new routing and forwarding features without changing the routers
  - today, most P2P overlay networks are built in the application layer on top of the TCP/IP networking suite
    - it means: P2P overlay networks are typically application level abstractions
• A P2P protocol defines the set of messages that the peers exchange, their format and their semantics.

• A common characteristic of all the P2P protocols:
  • identification of the peer through a **unique identifier**, which is generally computed through a hash function.
  • the protocols are defined at application level (stack TCP/IP) and define a routing strategy.
Unstructured overlays

- peers are arbitrarily connected
- the resulting logical network (overlay network) is unstructured
- look-up algorithms typically implemented:
  - out-of-band
  - flooding (Gnutella)
- Pro/Cons:
  - easy maintenance, diffusion
  - highly resilient
  - high lookup cost: linear in N, where N is the number of nodes in the network
  - low scalability

Examples

- Gnutella (up to v0.4)
- BitTorrent, P2P-TV systems (out-of-band lookup)
- BitCoin
UNSTRUCTURED OVERLAYS

- Peers directly interacts without the presence of a centralized server
- Interaction paradigm based on a decentralized cooperation, rather than a centralized coordinator.

Client/Server

Peer to Peer
HIERARCHICAL OVERLAYS

- Peers and super-peers
  - peer connect to Super-Peers
  - Super-Peers know Peer resources (index of the resources)

- Lookup
  - flooding restricted to Super-Peers
  - resources are directly exchanged between the peers.

- Pros/cons
  - lower lookup cost,
  - improved scalability
  - less resilient to Super-Peer churn

- Examples
  - Skype (for relay), eDonkey (pre Kad)
  - Gnutella (v0.6 on), Kazaa
STRUCTURED P2P OVERLAY

• The choice of the neighbours is defined according to a given criteria
  • the resulting overlay network is structured
• Goal: guaranteeing the la scalability
  • Look-up: the network structure guarantees that the look up of an information has a given complexity (for instance $O(\log N)$).
  • complexity guarantees also for peer joins and peer leave

• Examples.
  • distributed hash tables approaches
    • CAN, Chord, Pastry, Emule (Kademlia),...
TAXONOMY WITH RESPECT TO THE OVERLAY

Filesharing
- BitTorrent
- eDonkey

Live TV / VoD
- SopCast
- TVAnts
- PPLive
- Joost

VoIP/Chat
- Skype
- Gtalk

Search
- Kademlia

SW Updates
- RapidUpdate
- DebTorrent
- Apt-p2p,

- Structured
- Unstructured
- Hierarchical
P2P: ADVANTAGES

- Advantages; to exploit computational resources 'in excess' (idle CPU cycles, unused storage space, unused bandwidth,...) to have in return resources/services/social networks participation,..
- Advantages for the community
  - self scaling property: the participation of a larger number of users/host naturally increases the system resources and its capacity to serve a larger number of requests
- Advantages for the seller: decrease of the cost to set up a new application
  - client/server: requires the use of a server farm characterized by high connectivity so that the request of millions of users can be satisfied
  - fault tolerance: the server farm must be replicated in different locations
  - the server must be managed so to offer a service 24*7
P2P: THE FUTURE

• The success of a P2P application greatly depends from the creation of a "critical mass" of users: a level of participation to the P2P network which enables the application to self-sustain.

• In the first P2P systems the novelty of the application has been fundamental for reaching a critical mass:
  • file sharing: free content, wide content selection
  • VoP2P: low cost for international calls, good audio quality, ...
  • P2PTV: access to a wide set of programs, possibility of accessing the content from any location
  • cryptocurrencies as a profitable asset

• The success of new P2P applications will greatly depend besides a good engineering of the application:
  • from the new application appealing
  • from the definition of new 'business model'
Exploiting classical P2P techniques in different contexts:

- use of the blockchains in several fields:
  - Social networks
  - Distributed voting
  - IoT

- data-centers with thousand of machines exploit techniques taken from the P2P world.
  - Amazon Dynamo: distributed storage system over hosts belonging to the data center
    - highly scalable
    - simple: key/value put/get
    - guarantees service level agreement
    - exploits Distributed Hash Tables
    - trade-off availability/strong consistency

- P2P media streaming
SCIENTIFIC CHALLENGES

- The development of P2P applications is a challenge and requires the solution of several novel problems.
- The classic methodologies for the development of distributed systems of the “old generation” cannot be exploited:
  - the order of magnitude of a P2P system is different with respect to that of classical systems (million of nodes with respect to hundreds of nodes)
  - classical algorithms/techniques classiche “do not scale” on network of this size
  - the failure of one of the nodes is not a rare event, rather a normal event
- Systems with dimension and with this dynamicity level require new tools
  - probabilistic algorithms
  - game theory: strategies for the peer cooperation
  - cryptographic techniques
  - novel consensus algorithm