Network Research Project Proposal

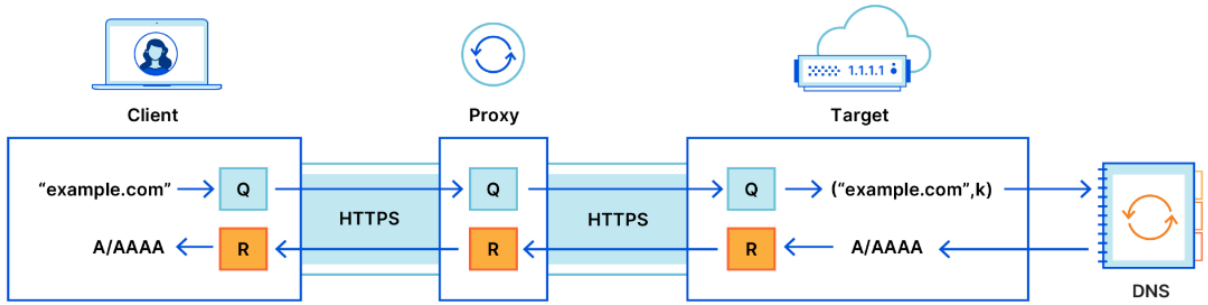
-- CS 397/497 Selected Topics in Computer Networks, Spring 2023

Yunming Xiao

yunming.xiao@u.northwestern.edu

Project 1: Non-Collusion In the Wild

- There is a new trend to enhance various Internet systems by operating with multiple parties and non-collusion agreement
  - Multi-party Relay
  - Oblivious DNS
  - Private Aggregate Statistics
  - Advertisement Auction
  - Privacy Pass
  - ……



- Despite the proliferation of theses systems, the non-collusion agreements which they rely on is barely studied

Project 1: Non-Collusion In the Wild

- This study aims at investigating the deployment and practice of non-collusion agreements in the current Internet

- Methodology
  - Parse the announcements and legal documents if available
  - Join the system as a user, and try to measure who is involved and how the user data is handled by different parties
  - ……

Project 1: Non-Collusion In the Wild

- Goals are to understand
  - What kind of systems are dependent on non-collusion agreement
  - What is exactly the non-collusion agreement about?
  - Who are involved in the non-collusion agreement
  - Limitation of the non-collusion agreements, e.g., geological limitation
  - Performance of these systems, compared to the ones without NCA
  - Is there any bad practice of the non-collusion agreement?
    - Example 1: A and B signed the NCA, yet B is building its service on AWS which does not have an NCA with either A or B
    - Example 2: B is a temporary company, which is directly or indirectly funded by A
    - Example 3: The NCA between A and B only covers part of the data but not all

Project 1: Non-Collusion In the Wild

- Higher Goals are to answer
  - What is the best practice/principle to execute non-collusion agreement?
  - Is there any technical solution that we can ensure the correct execution of NCA, and allow the users to be able to verify that NCA is correctly executed?
  - Is there any system that we can remove the NCA with more advanced cryptographies or through system redesign?

Project 2: A First Look at Decentralized Apps

- Decentralized applications (DApps) allows Internet users to share their bandwidth or computational resources with others, potentially in exchange for a monetary benefit
- The rise of cryptocurrencies based on blockchain has given rise the DApps because they allow convenient small transactions to be made between people across the globe
- Examples:
  - Tor, or Mysterium (DVPN)
  - Golem Network (CPU Resource)
  - Render Network (GPU Resource)
  - Filecoine (Memory Resource)
  - ……

Project 2: A First Look at Decentralized Apps

- Why interesting – DApps have ***different*** privacy properties from the systems relying on non-collusion agreement
  - DApps is **weaker** because no actual NCA is present, and hence less legal bound
  - Meanwhile, DApps is **stronger** because it scatters the data to much more parties, making it hard to re-centralize if needed

- Methodology
  - Join DApps as a user and hence perform active measurements
  - Join DApps as a provider and hence perform passive measurements
  - Build models to describe the systems

Project 2: A First Look at Decentralized Apps

- Goals are to better understand the DApps in terms of
  - Performance
  - Daily active volume
  - Security and privacy protection as well as reliability of the service
  - Risks of using the DApps or joining DApps as providers
  - The economic aspects of the DApps and the *resource marketplace* behind them

- Higher Goals are to answer
  - Is there any technical solution that we can ensure and also allow the users to verify that the DApps service providers do not harm the user privacy?

Project 3: Eliminate the Trust: The Case of Ad Exchange

- The best privacy-preserving system is the one that removes the dependency on the non-collusion agreement
- A good example is PDNS

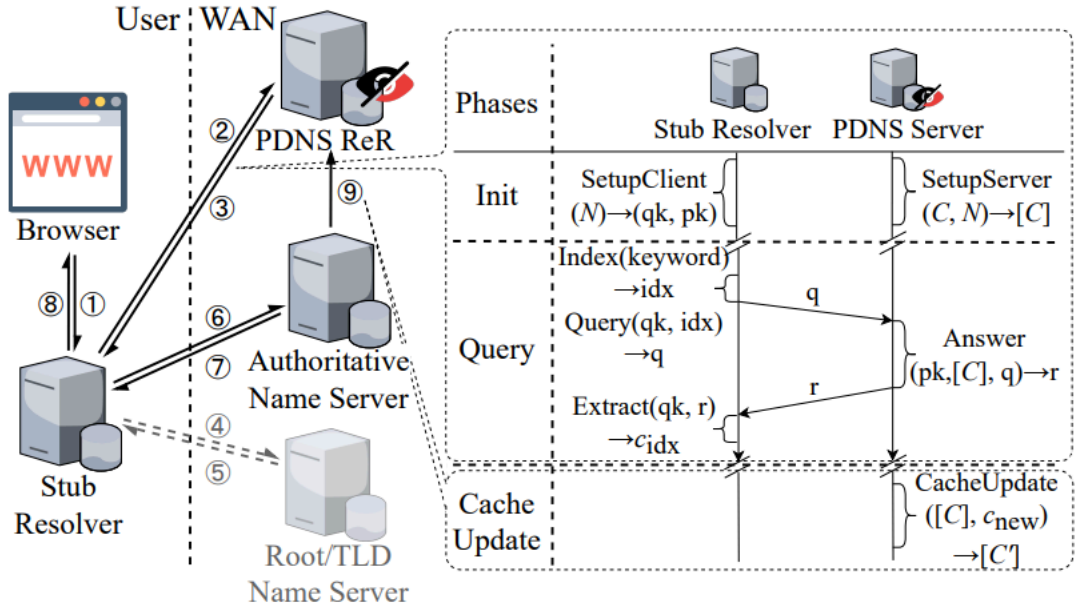


Figure 1: Visualization of PDNS and its workflow.

Project 3: Eliminate the Trust: The Case of Ad Exchange

- A good example is PDNS

Table 1: Comparison of privacy-preserving properties of current DNS solutions versus single and multi-server PIR.

Solution	Defend Pervasive Monitoring	Hide Individual Access Pattern	Hide Organizational or Regional Access Pattern	Survive Non-Collusion Agreement Violation
DoUDP [16] / DoTCP [43]	No	No	No	N/A
DoT [57] / DoH [54]	Yes	No	No	N/A
DoT/DoH + Resolver Rotation [56, 80]	Yes	Yes*	No	N/A
Oblivious DNS [82]	Yes	Yes	No	No
ODNS With Proxy Rotation [63]	Yes	Yes	Yes*	No
DoHoT [69, 70]	Yes	Yes	Yes*	Yes*
DNS with Multi-Server PIR	Yes	Yes	Yes	No
DNS with Single-Server PIR	Yes	Yes	Yes	Yes

Project 3: Eliminate the Trust: The Case of Ad Exchange

- Another example could be ad exchange

Figure 1: Online Advertising Auction

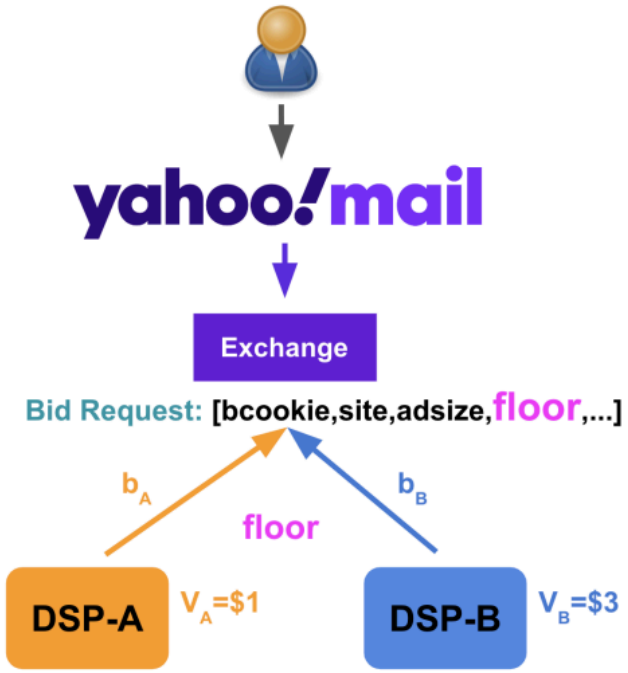


Figure helps to understand how bidding behavior of DSPs changes as a function of the auction mechanism and floors.

Project 3: Eliminate the Trust: The Case of Ad Exchange

- An improvement is to employ multi-party computing to hide the bidder's information
  - User privacy is not protected

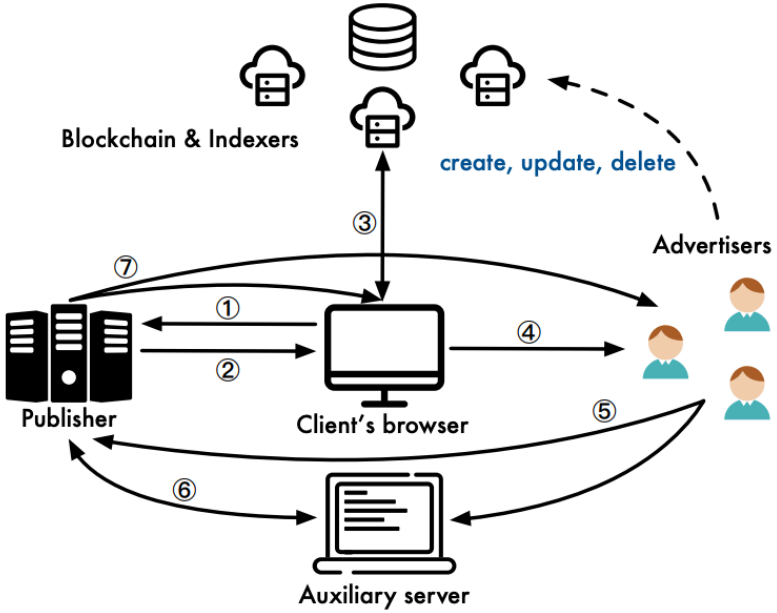
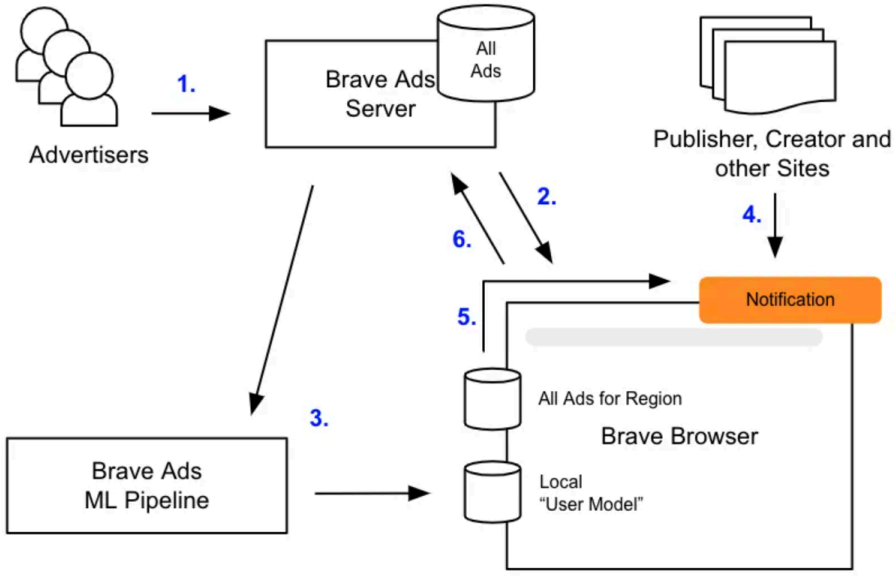


FIGURE 1—In Addax, the exchange's functionality is divided among the publisher, browser, an auxiliary server and a blockchain.

Project 3: Eliminate the Trust: The Case of Ad Exchange

- To protect user privacy, one solution is to not send it out to anyone but decides the ads to watch locally (Brave)
  - But here the ad provider cannot decide their auction schemes



Project 3: Eliminate the Trust: The Case of Ad Exchange

- Our approach
  - The data shouldn’t leave the users’ devices
  - We ask the ad providers to send their auction algorithm to the user, which performs the computation and auction
  - Some cryptography tools can ensure that the user does not understand the ad providers’ algorithm while allowing the ad provider to verify that the user did not cheat
  - The cost is it incurs much higher computation requirements

- Goals of this study:
  - Find out whether the above can be done, and how it could be done
  - Evaluate the overhead of the above approach

Sen Lin

sen.lin@u.northwestern.edu

Metaverse Streaming

#metaverse #streaming #vr #cloud-game #rust #cpp

Introduction

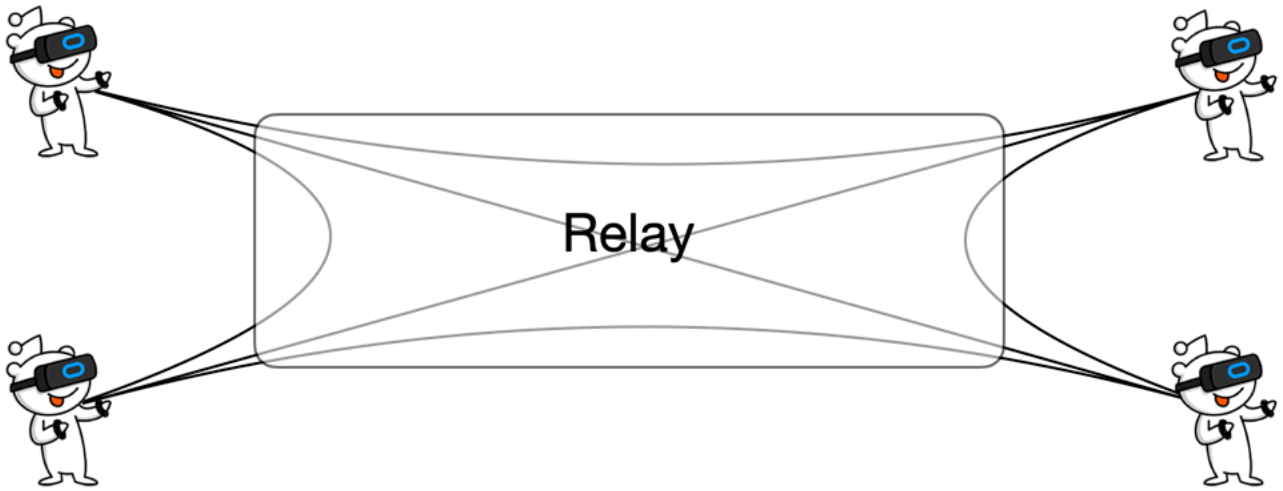
- The giant gap between the **expected metaverse experience** and **metaverse showcases**

Image credit: Meta

- The metaverse is design to enable “digital twins” between the physical world and the virtual world
- However, the player’s presence (what is **uploaded** from the client side) in the virtual world has fallen far short of the design goal
- Status quo: still the same level of experience as traditional MMO games

Introduction

- Issues from the perspective of computer networks
  - Higher upstream bandwidth and tighter latency requirements
  - The simple "*relay*" network design in traditional MMO games is not sufficient

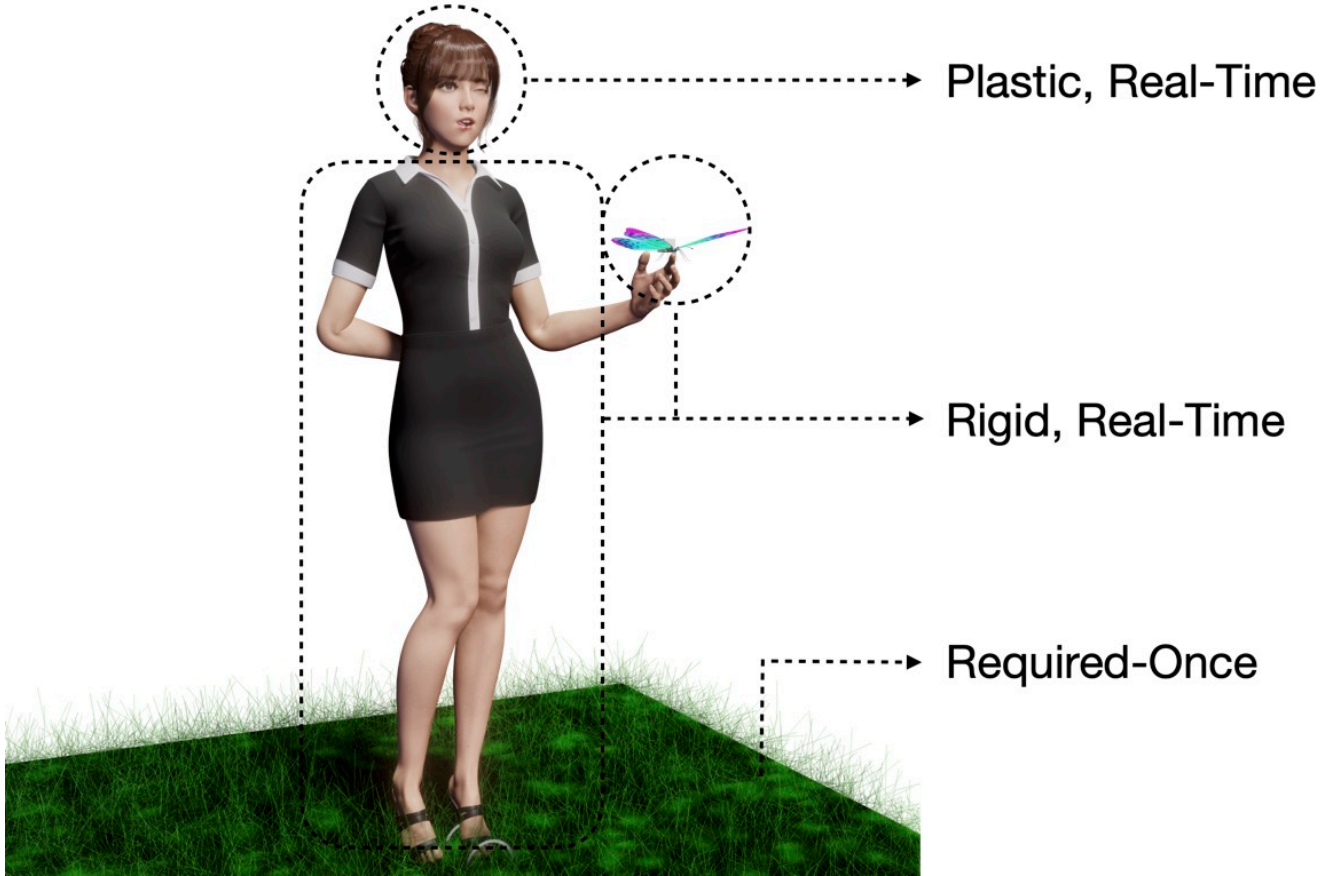


The bandwidth is linear to the number of players

Idea – Divided We Stand

Image credits: Cyberpunk 2077

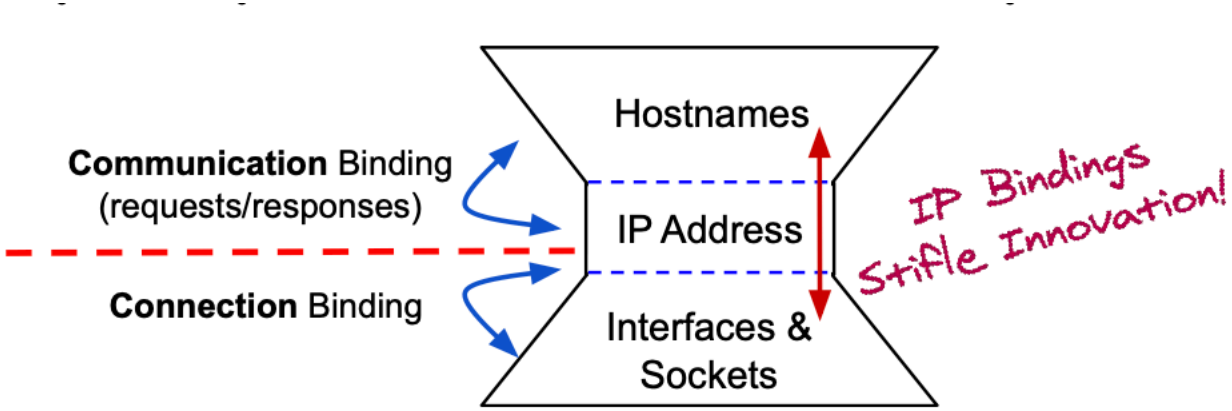
- Step 1: Evolve the "*relay*" network to "*edge-assisted*" networks
  - Ensure a constant bandwidth demand
- Step 2: Divide the traffic **types** in metaverse streaming
  - Prioritize critical traffic
  - Loose some reliability requirement
- Step 3: Divide the metaverse **content** in metaverse streaming

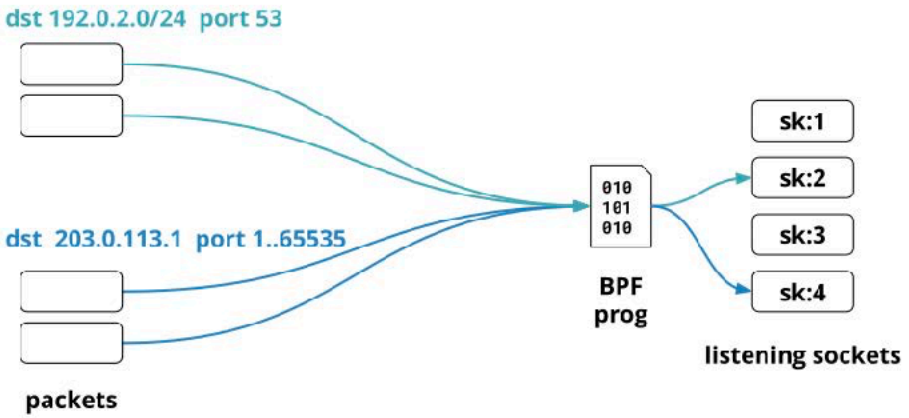


Fractional Connection

#transport #ebpf #rust #go

Motivation – Unbind IP (on the server side)





Recalling the relation between IP addresses, hostnames, and sockets, the IP address is bound to a connection because:

- It serves as parts of the connection identifier (as in TCP)
- System design strictly binds it to a socket

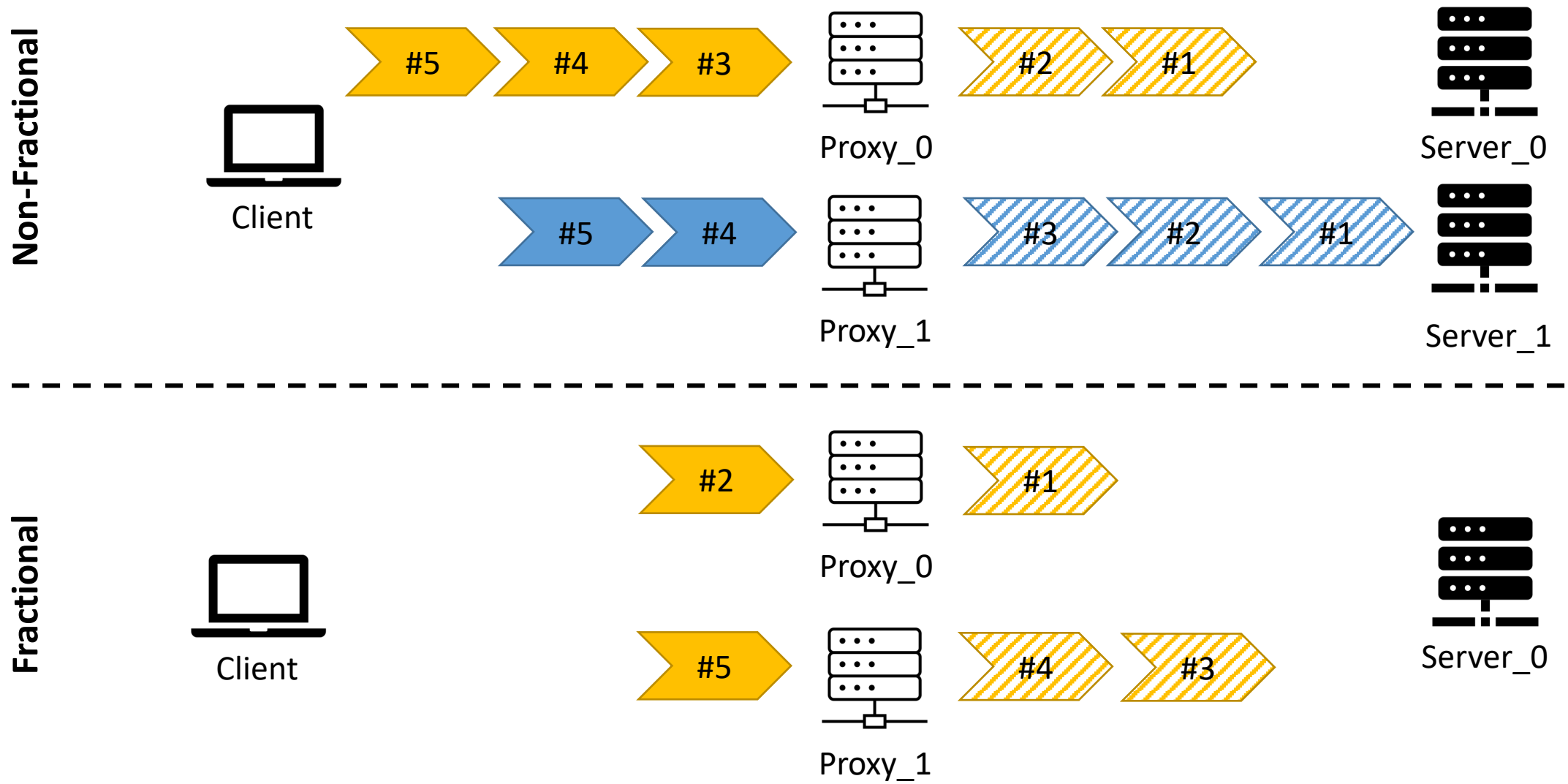
An **sk_lookup** example: Packets arriving on 192.0.2.0/24:53 to socket sk:2, while traffic on 203.0.113.1 to any port number lands in socket sk:4.

Credits of images in this slide: Fayed, Marwan, et al. "The Ties that un-Bind: Decoupling IP from web services and sockets for robust addressing agility at CDN-scale." Proceedings of the 2021 ACM SIGCOMM 2021 Conference. 2021.

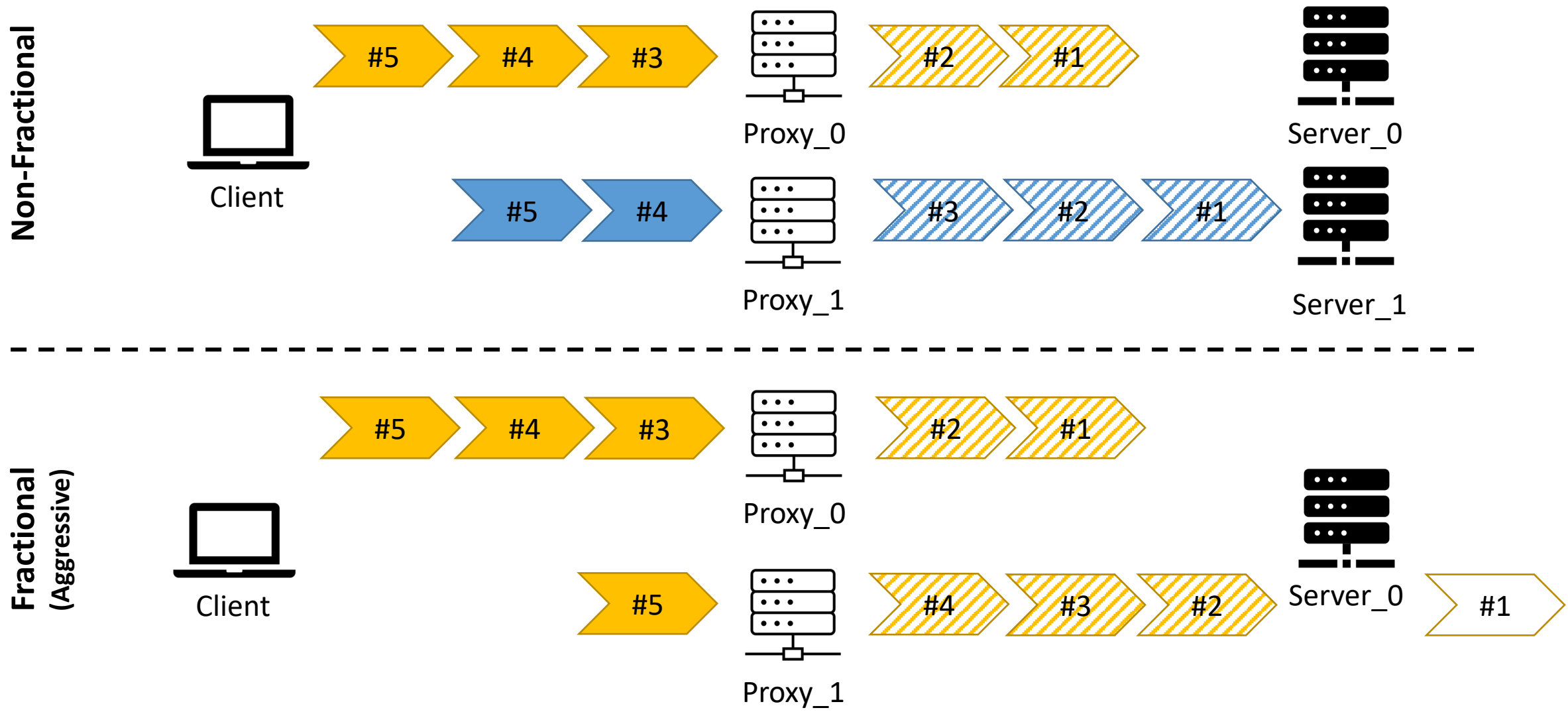
Idea – Unbind IP (on both sides)

Leverage the recently introduced IP-Socket unbinding technique. the IPs and ports are no longer associated with a “connection”, i.e., socket. We can maximize the use of socket mobility to fractionalize a connection. We assign a fixed connection ID/token in the packet header, e.g., TCP TS. Then the connection becomes fractional, which can be divided into multiple paths at the packet level.

Case study – Proxy/VPN



Case study – Proxy/VPN



Yanzhi Li

YanzhiLi2026@u.northwestern.edu

Decentralized Web

Yanzhi Li

Concept

The decentralized web, also known as the DWeb, refers to a vision for the internet where information, applications, and services are distributed across a network of computers without relying on a central authority or single points of control. The goal is to create a more resilient, open, and democratic infrastructure for the internet, in contrast to the current centralized web where power and control are concentrated among a few large entities.

Key Features

- Decentralized storage: Data is stored across multiple nodes, so there is no single point of failure or control. Examples of decentralized storage protocols include InterPlanetary File System (IPFS) and Filecoin.
- Decentralized applications (dApps): These applications run on decentralized platforms, like Ethereum or Cardano, that use smart contracts to enable decentralized functionality. dApps are typically open-source and do not rely on centralized servers or services.
- Peer-to-peer (P2P) networks: These networks allow computers to communicate and share resources directly with one another, without the need for a central server. Examples include BitTorrent and the Dat protocol.
- Decentralized identity: This approach to identity management allows users to control their own digital identities, without relying on centralized services or third-party providers. Examples include the Decentralized Identity Foundation (DIF) and projects using blockchain technology for identity verification.

Measurement Study

- Quantify DWeb's benefits and drawbacks, comparing with large scale web server like AWS.

  Privacy, Performance, Scalability, Robustness, etc.

- Propose prototype of DWeb applications

Start Point（IPFS）:

Design and Evaluation of IPFS:A Storage Layer for the Decentralized Web (acm.org)

Differential Privacy in DNS query

Yanzhi Li

Privacy Risks in DNS query

DNS queries are often sent unencrypted over the internet, making it easy for third parties, such as ISPs, governments, or malicious actors, to monitor and collect data about your browsing habits. This data can be used for targeted advertising, censorship, or even cyber attacks.

Profiling: By monitoring DNS queries, third parties can build detailed profiles about users' interests, preferences, and behavior. This information can be used for various purposes, such as targeted advertising, discrimination, or even more invasive surveillance.

Data leakage: Unencrypted DNS queries can reveal information about the devices and software you are using, your network configuration, and your location. This information can be exploited by malicious actors for targeted attacks or to exploit vulnerabilities in your devices or network.

Manipulation: DNS queries are susceptible to attacks like DNS spoofing or cache poisoning. In these attacks, an attacker can manipulate DNS responses, redirecting users to malicious websites or intercepting communications. These attacks can compromise users' privacy, as well as their security.

Differential Privacy

Differential privacy is a privacy-preserving mechanism that injects controlled noise into the data, so that the output of an analysis is still useful while providing a strong privacy guarantee for the individuals contributing to the data.

Basic Approach:

Local differential privacy: Each user's device would add noise to the DNS queries before they are sent to the DNS resolver. This approach ensures that the resolver cannot accurately determine the true query, but it can still provide a reasonably accurate response based on the noisy query.