Evaluating Power and Performance Consumption in a P2P Network While Mitigating a DDOS Attack Within ns-3

ENSC 427: Communication Networks
Spring 2025

Group 4

Fernando Arias 301435230 (fda7@sfu.ca)

Caleb Kuitenbrouwer 301398847 (cfk2@sfu.ca)

Clarence Wasilwa 301429747 (cww8@sfu.ca)

Outline

Introduction

- **Objective:**
  - Assess the impact of simulated DDoS attacks on a peer-to-peer network when mitigation is employed.
  - Measure performance degradation and increased power consumption during an attack.
- **Scope and Overview:**
  - Use ns-3 to simulate both normal P2P communications and a UDP flood DDoS attack.
  - Apply mitigation techniques to restore network performance and reduce energy usage.

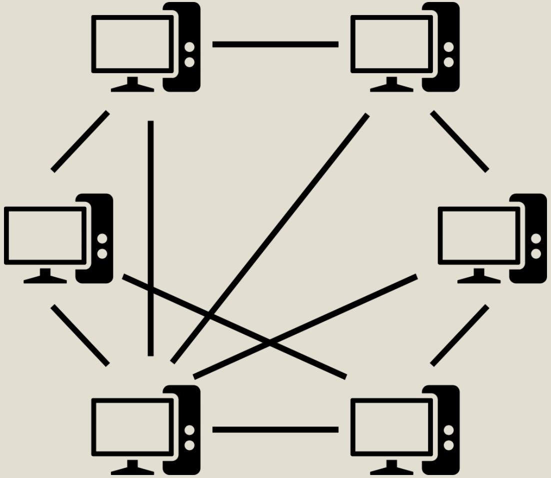
Motivation

- ➢ Thanks to the rise of Internet of Things (IoT) devices and decentralized applications, diverse Peer-to-Peer ecosystems have become more common.

- ➢ Peer-to-Peer network normalization could counter DDoS attackers, which is why it is important to address these network vulnerabilities and analyze the best methods to mitigate DDoS attacks. consumption even under attack..

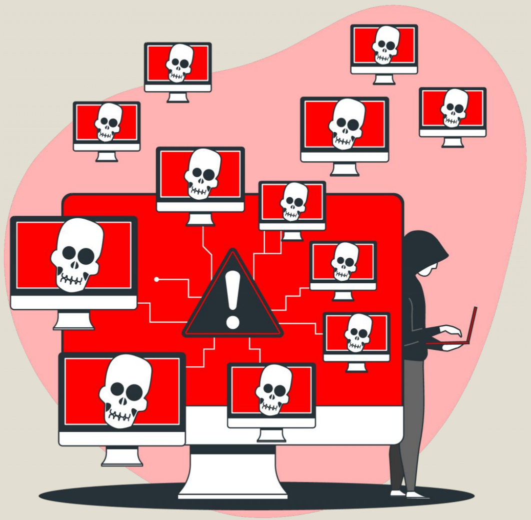
What is a P2P Network?

- A peer-to-peer (P2P) network is a decentralized system where every node can act as both a client and a server.
- It enables direct resource sharing among devices without relying on a central server.
- This approach increases resilience and can reduce bottlenecks.
- P2P networks are commonly used in file sharing, communication, and distributed computing.



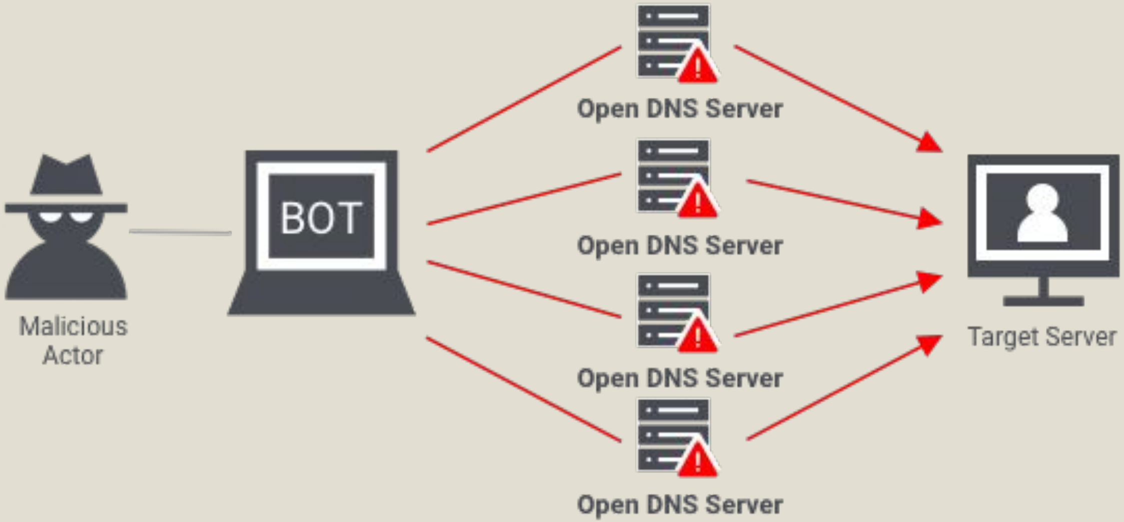
What is DDoS Attack?

- A Distributed Denial-of-Service (DDoS) attack involves multiple compromised systems flooding a target with overwhelming traffic.
- The objective is to exhaust the target's resources, rendering it inaccessible.
- Attackers coordinate these actions from various sources.
- There are many types of DDoS attacks. In this study we will focus on DNS Amplification DDoS Attack



What is DNS Amplification DDoS Attack?

A DNS resolver translates domain names into IP addresses, allowing users to access websites. In DDoS amplification attacks, attackers spoof the victim’s IP address in DNS queries, causing resolvers to send large responses to the victim, which overwhelms the network. DNS amplification typically ranges from 20 to 100 times the original query's size.

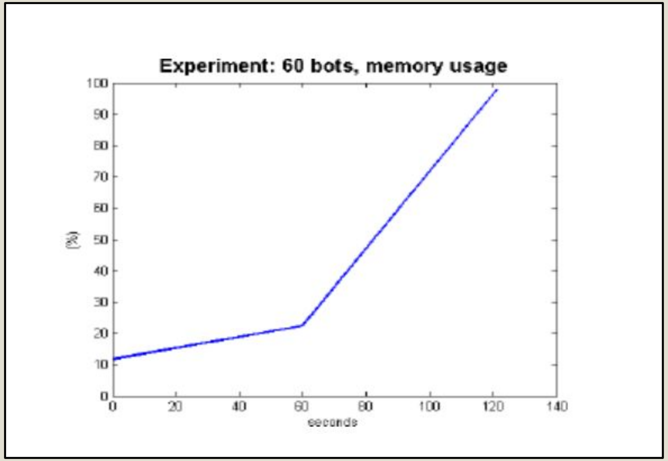


Overview of related Work

Simulation of DDoS Attacks on P2P Networks (By Nidal Qwasmi, Fayyaz Ahmed, and Ramiro Liscano):

- Developed a simulation framework that models UDP flood attacks on P2P networks.

- Demonstrated that DDoS attacks can degrade network performance by reducing node dynamism and increasing maintenance overhead.



Overview of related Work (Cont..)

DDoS Testbed Based on Peer-to-Peer Grid (By Marek Simon and Ladislav Huraj):

- Designed a testbed using the OurGrid environment to emulate realistic DDoS scenarios in a P2P grid.
- Coordinated attacks via a botnet-like architecture, incorporating defense mechanisms such as port hiding and congestion control.
- Offers practical insights into deploying mitigation strategies and evaluating both performance and power consumption impacts.

Simulation Details

Simulation Environment:

NS-3 discrete event simulation environment

NS-3 Modules Used:
Core, Network, Internet, CSMA, Applications, Mobility, NetAnim, Random Variable Stream, FlowMonitor.

Programming Language & Standard Libraries:
Written in C++

Simulation Features:

- Mobility configuration with a grid-based placement
- Buffer management with distinct queue sizes for peers versus attacker/DNS
- Flow monitoring and network animation for detailed performance and visualization analysis

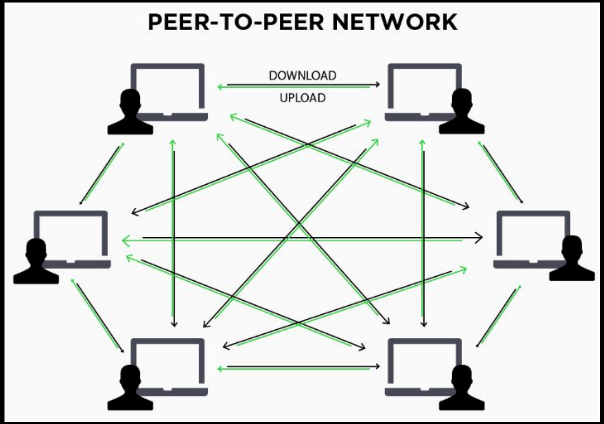
P2P Simulation Details

Network Topology:
5 peer nodes for regular data exchange with one main victim client that transmits at a higher rate.

Communication Channel:
All nodes connected via a CSMA channel with 100Mbps data rate and low latency (1ms).

Peers Behaviour:
Peers exchange data and acknowledgment (ACK) packets

Custom Traffic Elements:
Uses random delays to vary transmission times between nodes



P2P Code

Helper Functions:

```
*/
> void ClearRespondedPairs() ···

> /** ···
> void PacketReceived(Ptr<Socket> socket) ···

> /** ···
> void SendPacket(Ptr<Socket> socket, Ipv4Address destAddress, uint16_t port) ···

> /** ···
> void StartSending(Ptr<Socket> socket, std::vector<Ipv4Address> destAddresses, uint16_t port) ···
```

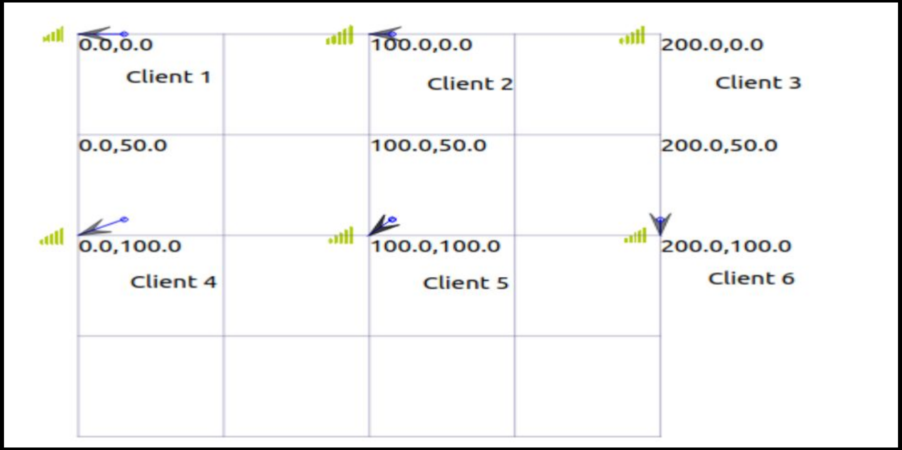
Code to create p2p:

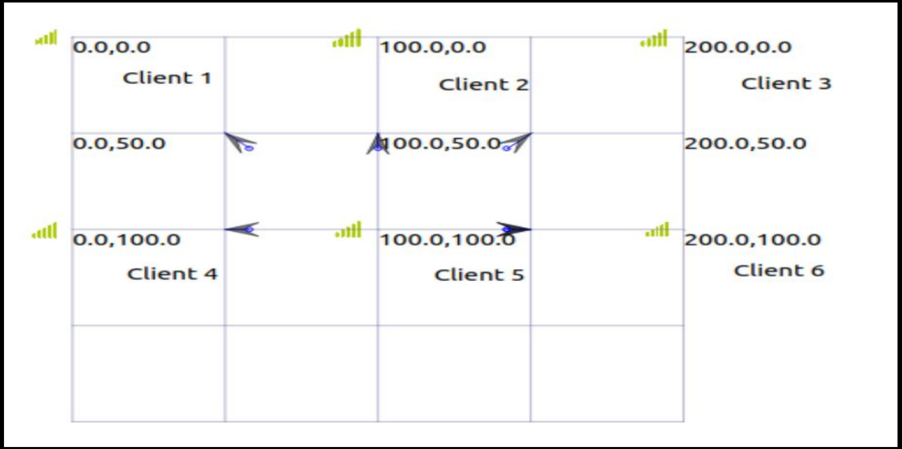
```
Ptr<Socket> s = Socket::CreateSocket(nodes.Get(i), UdpSocketFactory::GetTypeId());
s->Bind(InetSocketAddress(Ipv4Address::GetAny(), port));
s->SetRecvCallback(MakeCallback(&PacketReceived));
```

```
for (uint32_t i = 0; i < numPeers; ++i)
  Simulator::Schedule(Seconds(1.0), &StartSending, peerSockets[i], destAddressesForPeer[i], port);
```

P2P Animation

Using the NS3 mobility module and the netanim module we get the following animation of the simulated P2P network





DDoS Simulation Details

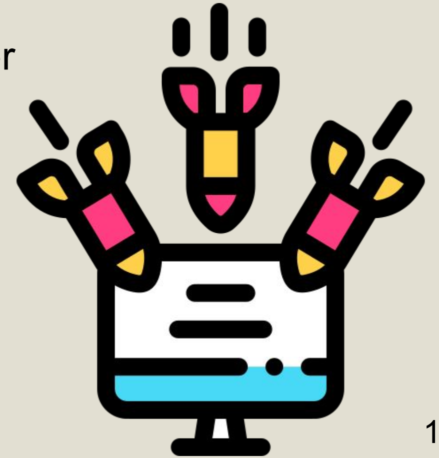
Attack Topology:
1 attacker node and 1 DNS resolver node

Communication Channel:
All nodes connected via a CSMA channel with 100Mbps data rate and low latency (1ms).

Node Roles:

- The attacker sends spoofed DNS queries
- The DNS resolver simulates response amplification toward the Main Peer

Custom Traffic Elements:
- Implements a custom header to simulate IP spoofing
- Implements an amplificated attack of 80 times the original query size



DDoS Code

Class for Spoofing

```
class SpoofHeader : public Header {
public:
  SpoofHeader() {}
  void SetSpoofedAddress(Ipv4Address address) { m_spoofedAddress = address; }
  Ipv4Address GetSpoofedAddress() const { return m_spoofedAddress; }

  static TypeId GetTypeId(void) {···
  virtual TypeId GetInstanceTypeId(void) const { return GetTypeId(); }
  virtual void Serialize(Buffer::Iterator start) const {···
  virtual uint32_t Deserialize(Buffer::Iterator start) {···
  virtual uint32_t GetSerializedSize(void) const {···
  virtual void Print(std::ostream &os) const {···
private:
  Ipv4Address m_spoofedAddress;
};
```

Scheduling Attack

```
// Schedule repeated spoofed DNS queries (the attack)
// The attacker spoofs the Main Peer's address (peerAddresses[0]) so that
// the amplified responses flood the Main Peer.
Simulator::Schedule(Seconds(10.0), &StartMultipleAttacks, attackerSocket,
```

Code to perform Amplification DDoS

```
> void SendSpoofedDNSQuery(Ptr<Socket> attackerSocket, Ptr<Socket> dnsSocket, Ipv4Address mainPeerAddress, uint16_t port)···

> /**···
void StartMultipleAttacks(Ptr<Socket> attackerSocket, Ptr<Socket> dnsSocket, Ipv4Address mainPeerAddress, uint16_t port)
{
  SendSpoofedDNSQuery(attackerSocket, dnsSocket, mainPeerAddress, port);
  Simulator::Schedule(Seconds(0.006), &StartMultipleAttacks, attackerSocket, dnsSocket, mainPeerAddress, port);
}
```

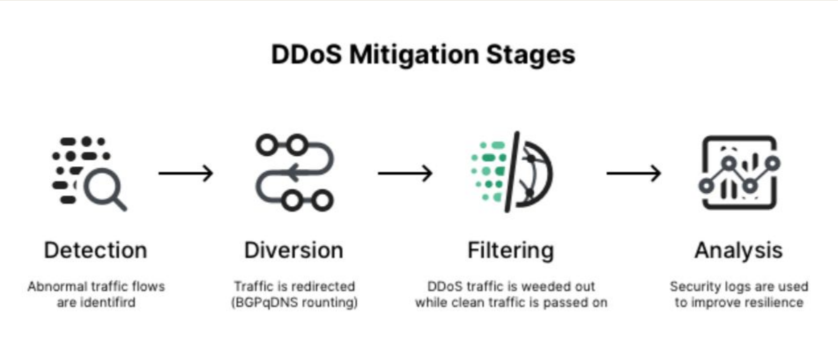
DDoS Mitigation Details

Mitigation Strategies Implemented:

- Timed Rate Limiter
- Blacklist IP Relegator

Timed Rate limiter:
Integrated into the victim nodes logic, detects if traffic exceeds a rate of over three received packets per second (3210 bps) from a singular IP address. If triggered the victim will subsequently drop further packets from the attackers IP until the rate decreases causing the victims internal clock to reset.

Blacklist IP Relegator:
Similar to the Time Rate Limiter the initial detection of burst packets received from an attacking node will be equivalent however further packets received from the selected IP will be immediately dropped.



Mitigation Code

Rate Limiter Mitigation

```
void PacketReceived(Ptr<Socket> socket)
{
  Address from;
  Ptr<Packet> packet = socket->RecvFrom(from);
  Ipv4Address senderAddress = InetSocketAddress::ConvertFrom(from).GetIpv4();
  std::string name = clientNames[socket];
  Time timestamp = Simulator::Now();
  if (name == "Main Peer")
    {
      if ((timestamp - lastResetTime).GetSeconds() >= 1.0)
      {
          packetCountMap.clear();
          lastResetTime = timestamp;
      }
      packetCountMap[senderAddress]++;
      //Over 10 from the same user sub second just never receive
      if (packetCountMap[senderAddress] > packetThreshold)
      {
          NS_LOG_WARN("[Mitigation] Main peer dropped the packet sent from " << senderAddress
                      << " at time " << timestamp.GetSeconds() << "s as its behaviour is suspicious");
          return;
      }
    }
```

Blacklist Mitigation

```
if (blacklist.find(senderAddress) != blacklist.end())
{
    NS_LOG_WARN("[Blacklist] " << name << " dropped a packet from "
                                << senderAddress << " at " << timestamp.GetSeconds() << "s");
    return;
}

if (name == "Main Peer")
{
    if ((timestamp - lastResetTime).GetSeconds() >= 1.0)
    {
        packetCountMap.clear();
        lastResetTime = timestamp;
    }
      packetCountMap[senderAddress]++;
    if (packetCountMap[senderAddress] > blacklistThreshold)
    {
        blacklist.insert(senderAddress);
        NS_LOG_WARN("[Mitigation] " << senderAddress << " blacklisted at "
                                     << timestamp.GetSeconds() << "s");
        return;
    }
```

Simulation Scenarios

The 4 situations analyzed in the simulation included:

1. Base Peer to Peer network.
2. Peer to peer network under an amplification DDoS attack.
3. Peer to Peer network under an amplification DDoS attack with Black List mitigation.
4. Peer to Peer network under an amplification DDoS attack with Rate Limiter mitigation.

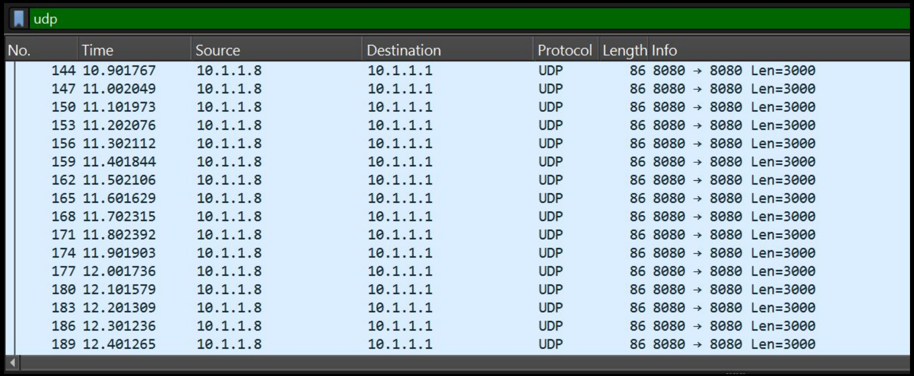
Results and Analysis

Wireshark analysis showing normal communication of the Peer to Peer Network

53	0.111321	10.1.1.6	10.1.1.4	UDP
54	0.113494	10.1.1.4	10.1.1.6	UDP
55	0.114930	10.1.1.1	10.1.1.6	UDP
56	0.114930	10.1.1.6	10.1.1.1	UDP
57	2.005608	10.1.1.6	10.1.1.3	UDP
58	2.009660	10.1.1.5	10.1.1.6	UDP
59	2.014066	10.1.1.6	10.1.1.1	UDP
60	2.029923	10.1.1.3	10.1.1.6	UDP
61	2.035564	10.1.1.6	10.1.1.2	UDP

1	0.000000	00:00:00_00:00:02	Broadcast	ARP	64	Who has 10.1.1.3? Tell 10.1.1.2
2	0.000754	00:00:00_00:00:04	Broadcast	ARP	64	Who has 10.1.1.5? Tell 10.1.1.4
3	0.003147	00:00:00_00:00:05	00:00:00_00:00:04	ARP	64	10.1.1.5 is at 00:00:00:00:00:05
4	0.003147	10.1.1.4	10.1.1.5	UDP	1070	8080 → 8080 Len=1024
5	0.005791	00:00:00_00:00:03	Broadcast	ARP	64	Who has 10.1.1.4? Tell 10.1.1.3
6	0.005791	00:00:00_00:00:04	00:00:00_00:00:03	ARP	64	10.1.1.4 is at 00:00:00:00:00:04
7	0.007956	00:00:00_00:00:06	Broadcast	ARP	64	Who has 10.1.1.2? Tell 10.1.1.6
8	0.009111	00:00:00_00:00:02	Broadcast	ARP	64	Who has 10.1.1.4? Tell 10.1.1.2
9	0.009111	00:00:00_00:00:04	00:00:00_00:00:02	ARP	64	10.1.1.4 is at 00:00:00:00:00:04
10	0.012664	00:00:00_00:00:04	Broadcast	ARP	64	Who has 10.1.1.1? Tell 10.1.1.4
11	0.014675	00:00:00_00:00:01	00:00:00_00:00:04	ARP	64	10.1.1.1 is at 00:00:00:00:00:01
12	0.014675	10.1.1.4	10.1.1.1	UDP	1070	8080 → 8080 Len=1024
13	0.016876	10.1.1.2	10.1.1.4	UDP	1070	8080 → 8080 Len=1024
14	0.017922	00:00:00_00:00:06	Broadcast	ARP	64	Who has 10.1.1.1? Tell 10.1.1.6
15	0.019876	00:00:00_00:00:04	Broadcast	ARP	64	Who has 10.1.1.2? Tell 10.1.1.4
16	0.019953	00:00:00_00:00:02	Broadcast	ARP	64	Who has 10.1.1.5? Tell 10.1.1.2

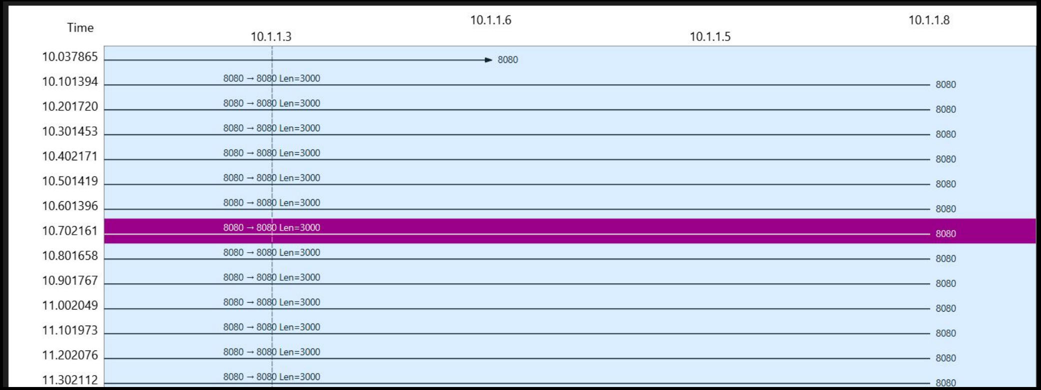
Results and Analysis (Cont..)



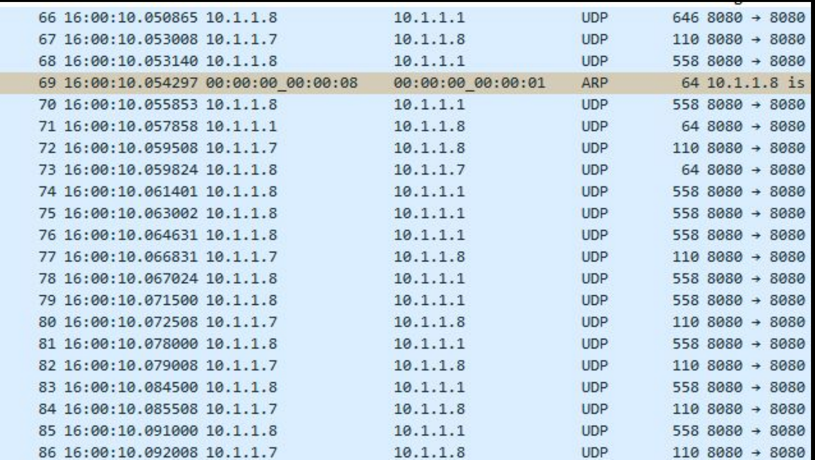
udp

No.	Time	Source	Destination	Protocol	Length	Info
144	10.901767	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
147	11.002049	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
150	11.101973	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
153	11.202076	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
156	11.302112	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
159	11.401844	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
162	11.502106	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
165	11.601629	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
168	11.702315	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
171	11.802392	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
174	11.901903	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
177	12.001736	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
180	12.101579	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
183	12.201309	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
186	12.301236	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000
189	12.401265	10.1.1.8	10.1.1.1	UDP	86	8080 → 8080 Len=3000

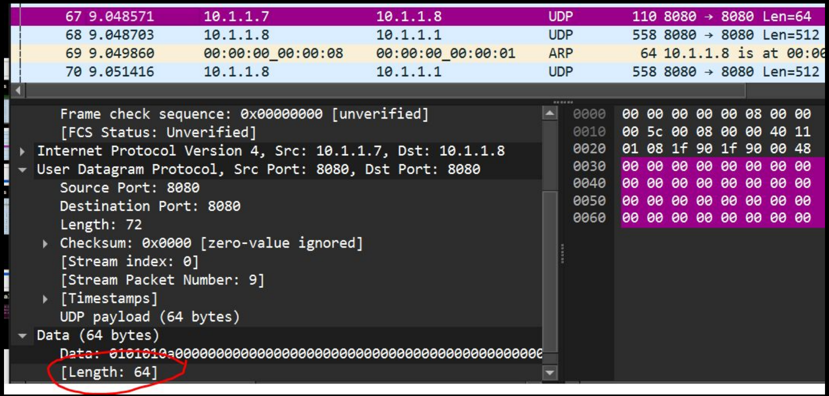
Wireshark flow graph showing the region of t = 10 seconds when DDoS attack started and victim (10.1.1.1) was bombarded by server (10.1.1.8)

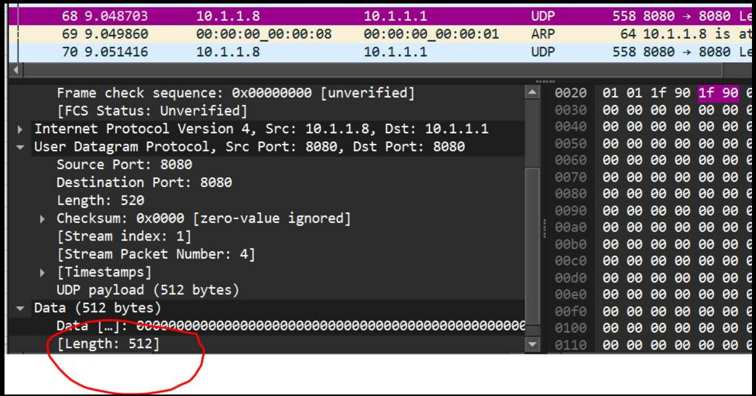


Results and Analysis (Cont..)

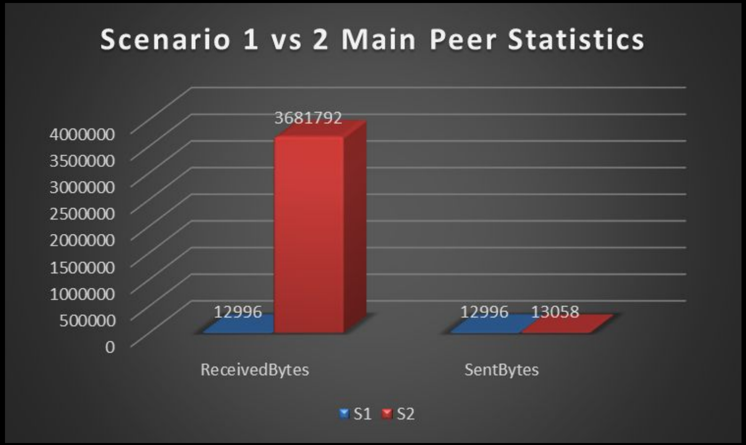


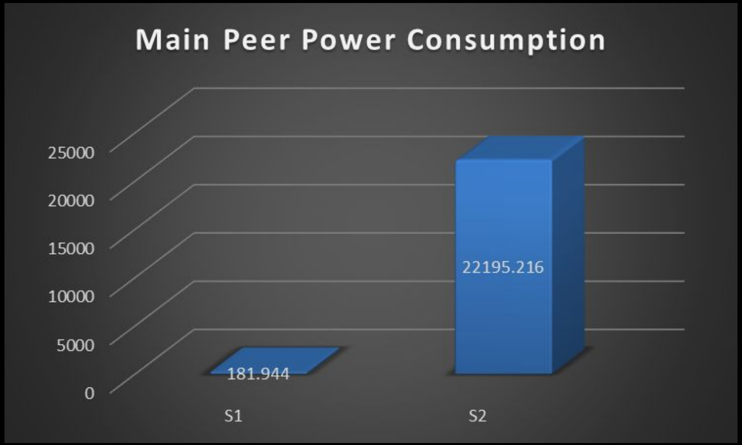
Wireshark analysis showing instances when the black list mitigation came into effect

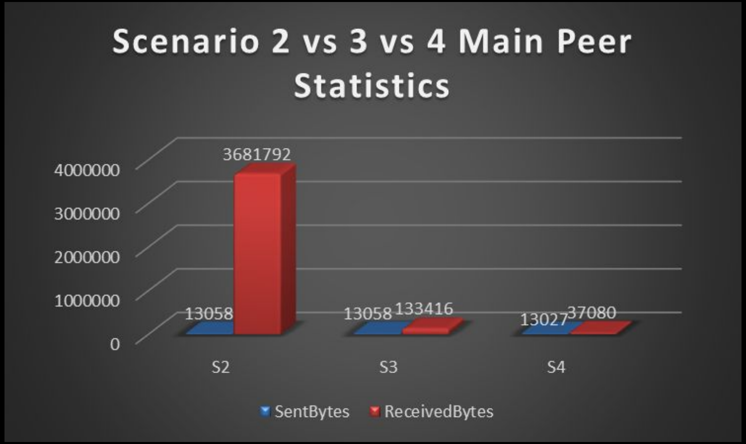


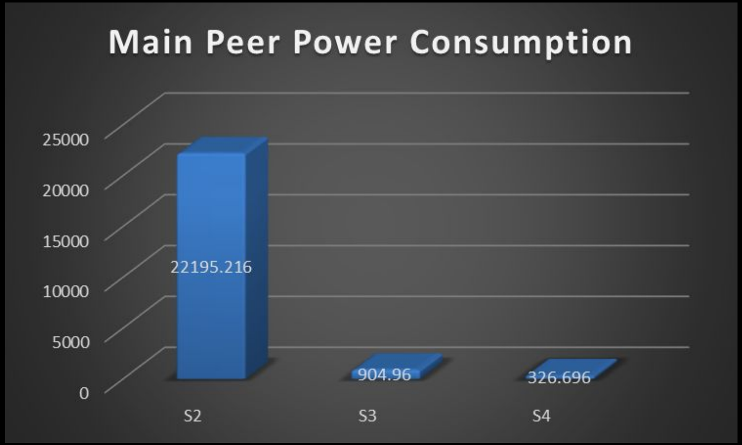


Results and Analysis (Cont..)

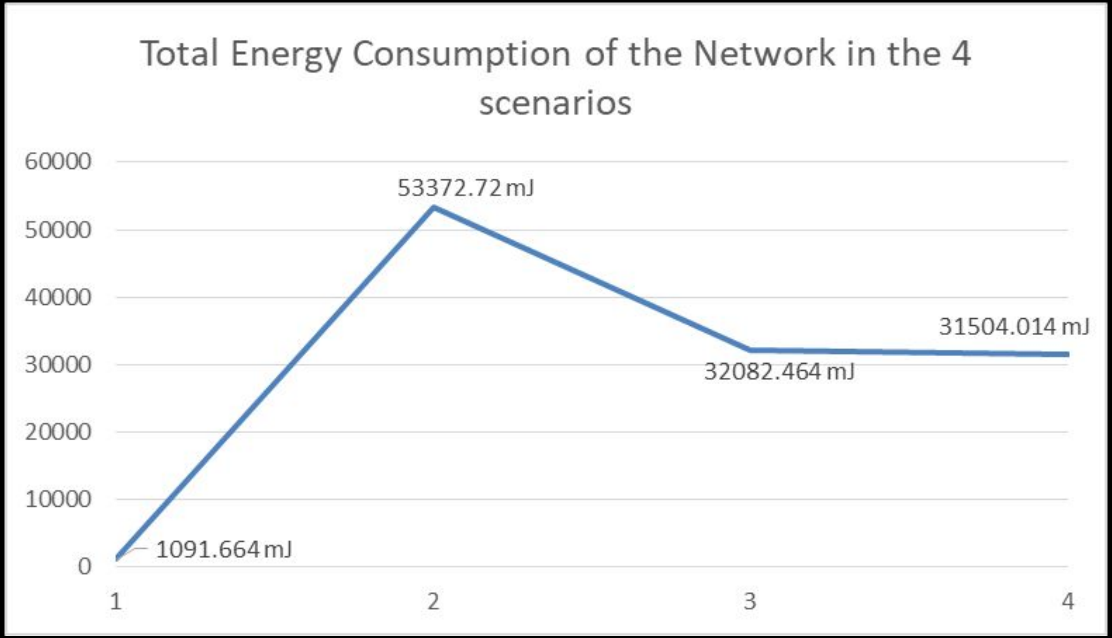








Results and Analysis (Cont..)



Logs

```
ofing Main Peer (10.1.1.1)
[Time: 10.052s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.1
)
[Time: 10.053s] DNS Resolver received data packet from 10.1.1.7
[Time: 10.0541s] Main Peer received data packet from 10.1.1.8
[Time: 10.0585s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, sp
oofing Main Peer (10.1.1.1)
[Time: 10.0585s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.
1)
[Time: 10.0597s] DNS Resolver received data packet from 10.1.1.7
[Mitigation] 10.1.1.8 blacklisted at 10.0639s
[Time: 10.065s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, spo
ofing Main Peer (10.1.1.1)
[Time: 10.065s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.1
)
[Time: 10.0665s] DNS Resolver received ACK from 10.1.1.1
[Time: 10.0715s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, sp
oofing Main Peer (10.1.1.1)
[Time: 10.0715s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.
1)
[Time: 10.0731s] DNS Resolver received data packet from 10.1.1.7
[Time: 10.0744s] DNS Resolver received data packet from 10.1.1.7
[Blacklist] Main Peer dropped a packet from 10.1.1.8 at 10.0775s
[Time: 10.078s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, spo
ofing Main Peer (10.1.1.1)
[Time: 10.078s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.1
)
[Blacklist] Attacker dropped a packet from 10.1.1.8 at 10.0787s
[Time: 10.0799s] DNS Resolver received data packet from 10.1.1.7
[Time: 10.0845s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, sp
oofing Main Peer (10.1.1.1)
```

Logs Analysis

P2P network / Packet Transfer

```
[Time: 6.05312s] Client 2 received ACK from 10.1.1.4
[Time: 6.05455s] Client 5 received data packet from 10.1.1.2
[Time: 6.05455s] Client 5 sent ACK to 10.1.1.2
[Time: 6.05556s] Client 2 received ACK from 10.1.1.5
```

DNS Amplification DDoS Attack

```
[Time: 10.0585s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, sp
oofing Main Peer (10.1.1.1)
[Time: 10.0585s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.
1)
```

Logs Analysis (Cont..)

Blacklist Mitigation

```
[Time: 10.0597s] DNS Resolver received data packet from 10.1.1.7
[Mitigation] 10.1.1.8 blacklisted at 10.0639s
```

Mitigation Effect

```
[Blacklist] Main Peer dropped a packet from 10.1.1.8 at 10.0775s
[Time: 10.078s] Attacker sent spoofed DNS query (60 bytes + header) to DNS Resolver, spo
ofing Main Peer (10.1.1.1)
[Time: 10.078s] DNS Resolver sent amplified response (6000 bytes) to Main Peer (10.1.1.1
```

Future Improvements

- The most significant improvement is to implement simultaneous packet transfer between nodes.
- DDoS attack on the network's essential nodes
- Detection of peers of the network
- Implement packet intercept and spoofing of the packet.
- Attack simultaneous peers using multiple DNS Resolvers.

Contribution

All the members contribute an equal third of portion to the slides of this presentation

References

[1] D. Wallace, 'Cyber Attack Cheat Sheet [Infographic]'. Accessed: Apr. 04, 2025. [Online]. Available: https://infographicjournal.com/cyber-attack-cheat-sheet/

[2] R. K. Aji, 'Kelebihan Dan Kekurangan Jaringan Peer to Peer', Culun Blog. Accessed: Apr. 03, 2025. [Online]. Available: http://culunid.blogspot.com/2018/12/kelebihan-jaringan-peer-to-peer.html

[3] 'What is UDP Flood DDoS Attack? Definition & Protection⚔️'. Accessed: Apr. 03, 2025. [Online]. Available: https://www.wallarm.com/what/udp-flood-attack

[4] 'A Survey of Peer-to-Peer Network Security Issues'. Accessed: Apr. 04, 2025. [Online]. Available: https://www.cse.wustl.edu/~jain/cse571-07/ftp/p2p/

[5] 'What is DDoS mitigation?', Tree Web Solutions. Accessed: Apr. 04, 2025. [Online]. Available: https://treewebsolutions.com//articles/what-is-ddos-mitigation-61

[6] Šimon, Marek and Ladislav Huraj. "DDoS testbed based on peer-to-peer grid." 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES) (2016): 1181-1186.

[7] N. Qwasmi, F. Ahmed and R. Liscano, "Simulation of DDOS Attacks on P2P Networks," 2011 IEEE International Conference on High Performance Computing and Communications, Banff, AB, Canada, 2011, pp. 610-614, doi: 10.1109/HPCC.2011.86. keywords: {Computer crime;Peer to peer computing;Authentication;Servers;Protocols;Conferences;P2P;DDOS;distributed computing;security},

Questions?

SFU