Contents

1 Introduction

In today's dynamic and interconnected world, communication networks form the backbone of digital infrastructure, enabling seamless data exchange across diverse platforms and applications. As these networks evolve, understanding traffic characteristics has emerged as a pivotal factor in optimizing performance, bolstering security, and improving user satisfaction. This final project report investigates the multifaceted nature of network traffic, with a special emphasis on the classification of encrypted traffic—an increasingly vital domain given the widespread adoption of encryption protocols in modern systems. The project is organized into three cohesive parts, each contributing to a holistic exploration of theoretical foundations, contemporary research, and practical implications.

Part I tackles five essential networking questions that illuminate the core principles and challenges of data transmission. These questions span critical topics, including the diagnosis of slow file transfers, the mechanics of TCP flow control, routing strategies in multi-path networks, the advantages of Multipath TCP (MPTCP), and the underlying causes of packet loss. By addressing these issues, Part I establishes a robust theoretical groundwork that supports the analyses in the subsequent sections.

Part II offers in-depth summaries of three pioneering research papers on traffic classification, each advancing the field of encrypted traffic analysis. These works introduce innovative techniques—ranging from transforming flow data into images for convolutional neural network (CNN) classification, to leveraging hybrid classifiers for early identification in Encrypted ClientHello (ECH) contexts, to passively extracting user attributes from HTTPS traffic. A comparative analysis across these papers underscores their collective contributions, methodological diversity, and potential avenues for future research.

Part III, slated for future completion, will feature a practical investigation of network traffic using Wireshark. This hands-on analysis will involve capturing traffic from various applications, examining key attributes such as packet sizes and inter-arrival times, and evaluating privacy vulnerabilities through attacker scenarios. Upon completion, the report will conclude with a synthesis of findings from all three parts, delivering a unified perspective on network traffic behavior and its broader implications.

This report comprehensively details Parts I and II, providing an extensive examination of theoretical concepts and research-driven insights into

encrypted traffic classification. The forthcoming Part III will bridge the gap between theory and practice, culminating in a thorough understanding of traffic characteristics and their significance in the realm of communication networks.

2 Part I: General Questions

2.1 Slow File Transfer Analysis

When a user experiences slow file transfers, several factors at the transport layer could be responsible. Bandwidth limitations often play a significant role; for instance, attempting to upload a large file over a low-bandwidth connection can lead to prolonged transfer times. High latency, which may result from long distances or multiple network hops, can delay packet acknowledgments, particularly in TCP-based protocols like FTP. Packet loss, whether due to network errors or congestion, necessitates retransmissions, further slowing the process. Network congestion itself can cause queuing delays and reduced throughput, especially in shared environments. Additionally, incorrect MTU settings can lead to packet fragmentation, increasing overhead and reducing efficiency. To troubleshoot these issues, administrators can use tools such as `ping` to measure latency and packet loss, `traceroute` to identify problematic routes, and Wireshark to analyze TCP behavior, including window sizes and retransmissions. Ensuring adequate buffer sizes and checking for device-level congestion are also crucial steps.

2.2 TCP Flow Control Mechanism

TCP employs a sliding window mechanism for flow control, ensuring that the sender's data transmission rate aligns with the receiver's processing capacity. The receiver advertises its available buffer space, and the sender adjusts its window size accordingly. In scenarios where the sender has significantly higher processing power than the receiver, the receiver's buffer may become overwhelmed, leading to packet drops. This forces the sender to wait for acknowledgments, reducing overall throughput. While this mechanism prevents data loss, it can limit performance in cases of mismatched capabilities. Window scaling can mitigate some of these limitations by allowing larger buffer sizes, but the fundamental challenge of balancing sender and receiver capacities remains. This highlights TCP's design focus on reliability over raw speed, dynamically adapting to network conditions.

2.3 Routing in Multi-Path Networks

In networks with multiple paths between source and destination, routing decisions critically impact performance. Path selection influences latency, with fewer hops generally reducing transmission delay. Throughput is affected by the bandwidth of the chosen path, as higher-capacity links can handle larger data volumes. Reliability also plays a key role, with stable paths minimizing the risk of packet loss or disruptions. Routing protocols like OSPF use cost metrics, such as bandwidth, to optimize path selection, while BGP incorporates policy-based decisions. Factors such as hop count, available bandwidth, current congestion levels, and link stability must be carefully considered to ensure efficient data delivery, minimizing both latency and packet loss in dynamic network environments.

2.4 MPTCP Performance Improvement

Multipath TCP (MPTCP) enhances network performance by leveraging multiple paths simultaneously. It improves load balancing by distributing traffic across available links, optimizing bandwidth utilization. Resilience is another significant benefit, as MPTCP can seamlessly switch to alternate paths if one fails, ensuring continuous connectivity. Additionally, it increases throughput by aggregating bandwidth from multiple connections, such as combining Wi-Fi and cellular networks on a mobile device for faster data transfers. This makes MPTCP particularly valuable in heterogeneous or unreliable network settings, offering both flexibility and an improved user experience.

2.5 Packet Loss Analysis

High packet loss between routers can arise from issues at both the network and transport layers. Network-layer causes include congestion, router buffer overflow, or misconfigured Quality of Service (QoS) policies, all of which can lead to dropped packets. At the transport layer, TCP retransmission timeouts or UDP's inherent lack of reliability can exacerbate the impact of lost packets. To address these issues, administrators should review router logs for errors, adjust QoS settings to prioritize critical traffic, and increase buffer sizes to handle traffic bursts. Tools like Wireshark can help identify loss patterns, enabling targeted interventions to restore reliable data transmission.

3 Part II: Research Papers

3.1 FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition

The paper *FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition* introduces a novel method for classifying encrypted network traffic, such as VPN and Tor, by transforming flow data into visual representations called FlowPics. These FlowPics are two-dimensional histograms where the x-axis represents normalized packet arrival times and the y-axis represents packet sizes. By applying Convolutional Neural Networks (CNNs) to these images, the method achieves high classification accuracy without the need for manual feature extraction. The approach is particularly innovative because it leverages the pattern-recognition capabilities of CNNs, traditionally used in image processing, to identify traffic categories and applications based solely on packet sizes and arrival times.

The traffic features used in this method are packet sizes and packet arrival times, transformed into a 1500x1500 image matrix. This image-based representation is a key novelty, enabling the CNN to automatically learn and extract relevant features from the visual patterns in the FlowPics. Evaluated on the UNB ISCX datasets, the method achieves over 96% accuracy for most traffic categories and over 99% for VPN traffic, even when trained on non-VPN data. It also demonstrates robustness to new applications, with 99.9% accuracy for classifying Facebook video traffic when the model was not specifically trained on that application. The approach performs well on encrypted traffic traversing VPN and Tor, making it versatile for real-world scenarios. With minimal storage requirements and potential for real-time classification, FlowPic outperforms previous methods and sets a new standard for encrypted traffic classification.

3.2 Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study

The study *Early Traffic Classification With Encrypted ClientHello: A Multi-Country Study* presents hRFTC, a hybrid Random Forest Traffic Classifier designed for early traffic classification (eTC) in the context of Encrypted ClientHello (ECH). As encryption techniques like ECH obscure critical metadata such as the Server Name Indication (SNI), traditional classification

methods struggle. To address this, hRFTC combines unencrypted TLS metadata from ClientHello and ServerHello messages (e.g., Cipher Suites length, Extension types) with flow-based statistical features extracted before the arrival of application data. These statistical features include packet size (PS) and Inter-Packet Time (IPT) metrics such as mean, standard deviation, and histograms. The hybrid approach is enhanced by a novel packet selection criterion that minimizes classification delay.

The traffic features used are a combination of TLS metadata and flow-based statistics, a novel strategy for early classification in ECH scenarios. Evaluated on a large, diverse dataset collected from multiple countries, the method achieves up to 94.6% F-score, significantly outperforming state-of-the-art packet-based, flow-based, and hybrid algorithms. In contrast, packet-based algorithms relying solely on TLS features perform poorly in ECH scenarios, with F-scores as low as 38.4%. The study underscores the importance of integrating flow-based features with TLS metadata for robust early classification. Additionally, it highlights the need for retraining hybrid algorithms in different geographic locations due to variations in traffic patterns, offering valuable insights for practical deployment.

3.3 Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application

The paper *Analyzing HTTPS Encrypted Traffic to Identify User's Operating System, Browser and Application* demonstrates that passive analysis of HTTPS encrypted traffic can reveal sensitive user attributes, such as the operating system (OS), browser, and application in use. The researchers introduce new features capturing browser-specific behaviors and SSL/TLS characteristics, supplemented by a large dataset of labeled HTTPS sessions. The method combines traditional base features—such as packet counts, bytes transferred, inter-arrival times, and packet size statistics—with innovative features like bursty behavior and SSL details. Bursty behavior features reflect traffic "peaks" and "bursts" tied to browser activities (e.g., loading multiple web page elements simultaneously), while SSL features include handshake details like SSL version, cipher methods, and extension counts.

The novelty of this approach lies in the introduction of bursty behavior and SSL-related features tailored for identifying user attributes from en-

crypted HTTPS traffic. The results are compelling: the method achieves 96.06% accuracy in classifying the ¡OS, Browser, Application¿ tuple, with near-perfect identification of OS and browser types. The inclusion of new features significantly boosts performance compared to using only baseline features. However, the study also raises privacy concerns, as it shows that even encrypted traffic can leak information about user devices and applications. This underscores the need for stronger privacy protections while highlighting the effectiveness of feature engineering in traffic analysis.

3.4 Cross-Paper Insights

The three papers collectively address the challenge of classifying encrypted network traffic, a task complicated by encryption techniques that obscure traditional payload-based features. Each paper offers a unique perspective, contributing innovative methods and insights that advance the field of traffic classification.

All three studies tackle the difficulty of classifying traffic when encryption hides packet payloads. *FlowPic* uses packet sizes and arrival times, which remain accessible despite encryption. The second paper addresses the specific challenge of Encrypted ClientHello (ECH), which obscures TLS metadata, by leveraging unencrypted handshake data and flow statistics. The third paper shows that HTTPS traffic reveals user attributes through behavioral and protocol patterns, even when encrypted.

The papers employ distinct strategies to overcome encryption barriers. *FlowPic*'s image-based method transforms flow data into visual FlowPics processed by CNNs, eliminating manual feature engineering. The second paper uses a hybrid Random Forest approach, combining protocol metadata with statistical features for early classification. The third paper relies on feature engineering, introducing bursty behavior and SSL features to identify user attributes. These varied methodologies highlight the adaptability of machine learning and feature design in encrypted contexts.

Each paper introduces novel features. *FlowPic*'s image transformation of packet sizes and arrival times enables CNN-based classification. The second paper's combination of TLS metadata and flow statistics is critical for early classification in ECH scenarios. The third paper's bursty behavior and SSL features capture browser-specific patterns, offering a new way to analyze encrypted traffic. These innovations demonstrate the importance of creative feature selection in overcoming encryption challenges.

The papers achieve high accuracy in different contexts. *FlowPic*'s 96%+ accuracy excels in classifying traffic categories and applications, with robustness to new scenarios. The second paper's 94.6% F-score is optimized for early classification, vital for real-time applications. The third paper's 96.06% accuracy in identifying user attributes highlights its precision in specific tasks. Together, these results prove that encrypted traffic can be effectively classified across diverse use cases.

These studies suggest several directions for advancement. *FlowPic*'s image-based approach could be enhanced with advanced CNNs or combined with metadata. The second paper's geographic variability findings call for adaptive models. The third paper's privacy concerns and feature success encourage exploration of privacy-preserving methods and broader feature applications. Future work could integrate these approaches—e.g., combining image-based and metadata-driven techniques—or address evolving standards like QUIC and TLS 1.3.

In summary, these papers collectively enhance our understanding of encrypted traffic classification. Their innovative features, diverse methods, and strong results provide a comprehensive view of the field, paving the way for future research and practical solutions in network analysis and security.

Part III: Practical Experiment & Analysis

3.1 Required Packages and Tools

To run this project successfully, the following dependencies are required:

Python Packages (As in requirements.txt)

- **scapy** - For reading and analyzing PCAP files.
- **pandas** - For data processing and analysis.
- **matplotlib** - For generating visualizations.
- **seaborn** - For statistical graph plotting.
- **numpy** - For numerical computations.

Applications & Extensions Used

- **Wireshark** - To capture network traffic as PCAP files.
- **Python 3.10+** - Required for running the scripts.
- **Matplotlib & Seaborn** - Used for graphical representation of network behaviors.

3.2 Understanding `part3_2.py`

The script `part3_2.py` is responsible for processing network traffic and visualizing its characteristics. It performs the following operations:

1. **Reads PCAP Files:**
   - Extracts information such as packet timestamps, sizes, and flow identifiers.
2. **Processes Key Traffic Metrics:**
   - Computes Inter-Arrival Time (time gap between consecutive packets).

- Aggregates Packet Size Distribution (how different packet sizes vary).
- Calculates Flow Metrics (total flows, packet counts, and bytes transmitted).

3. **Generates Comparative Graphs:**
   - Inter-Arrival Time Distribution: Shows the frequency of packet time differences.
   - Packet Size Distribution: Highlights the variance of packet sizes.
   - Flow Metrics Comparison: Displays unique flows, packet totals, and byte totals.

3.3 Insights from the Graphs

The graphs allow us to distinguish different applications based on their network behavior:

- **Browsing Behavior Differences (Web1: Edge vs. Web2: Chrome)**
  - Chrome (web2.pcap) sends more bursty traffic compared to Edge (web1.pcap).
  - Edge (web1.pcap) has more regular intervals due to its request-response nature.
- **Streaming (YouTube) vs. Browsing (Web Surfing)**
  - Streaming generates steady, large data transfers, while browsing has more short-lived, bursty flows.
  - Packet sizes for streaming are consistently larger, while browsing has variable-sized packets.
- **Video Conferencing (Zoom) vs. Other Activities**
  - Zoom (zoom.pcap) has smaller but frequent packets for real-time communication.
  - Browsing and streaming have less frequent, but larger packets.

This analysis demonstrates that even without decryption, traffic classification can help identify applications based on behavior.

3.4 Traffic Analysis from an Attacker's Perspective

Suppose an attacker wants to identify which applications or websites a user is visiting. Two attack scenarios arise:

1. **Attacker Knows Each Packet's 4-Tuple Hash:**
   - If an attacker has access to **source IP, destination IP, source port, and destination port hashes**, they can infer application type based on **flow characteristics**.
   - Even **encrypted** HTTPS traffic follows distinct packet patterns.
2. **Attacker Knows Only Packet Size & Timestamp:**
   - Even without IP addresses, just knowing **packet sizes and timestamps** can reveal application type.
   - **Streaming vs. Browsing vs. Conferencing** produce different data flow rates.
   - **Web browsing** exhibits bursty traffic, while **video calls** show small, frequent packets.

Mitigation Strategies:

- Using **VPNs** and **Tor networks** can obscure flow metadata.
- Padding packets to uniform sizes reduces fingerprinting risk.
- Randomizing inter-arrival times limits behavior-based classification.

4. Behavioral Differences & Application in Traffic Analysis

Understanding the differences between **various applications' network behaviors** allows us to detect and classify them **even when traffic is encrypted**. Below are key insights:

Application Type	Packet Size Pattern	Inter-Arrival Times	Flow Behavior
Web Browsing (Chrome/Edge)	Mixed, bursty small-medium packets	Irregular bursty patterns

| Short-lived, bursty |
Music Streaming (Spotify)	Medium packets, periodic	Regular interval packets	Continuous, stable
Video Streaming (YouTube)	Large packets, high throughput	Steady flow	Long-lived, high data transfer
Video Conferencing (Zoom)	Small packets, high frequency	Low delay, frequent	Constant, real-time packets

Why This Matters for Part 3.4:

- Even **fully encrypted traffic** can be **classified by behavior.**
- Attackers can infer applications **without needing to decrypt** packets.
- **Machine learning models** can be trained to detect applications based on flow metrics.
- Behavioral analysis is used in **network security**, **traffic shaping**, and **cyber threat intelligence**.

5. Conclusion

This project explored traffic classification using both theoretical and practical approaches.

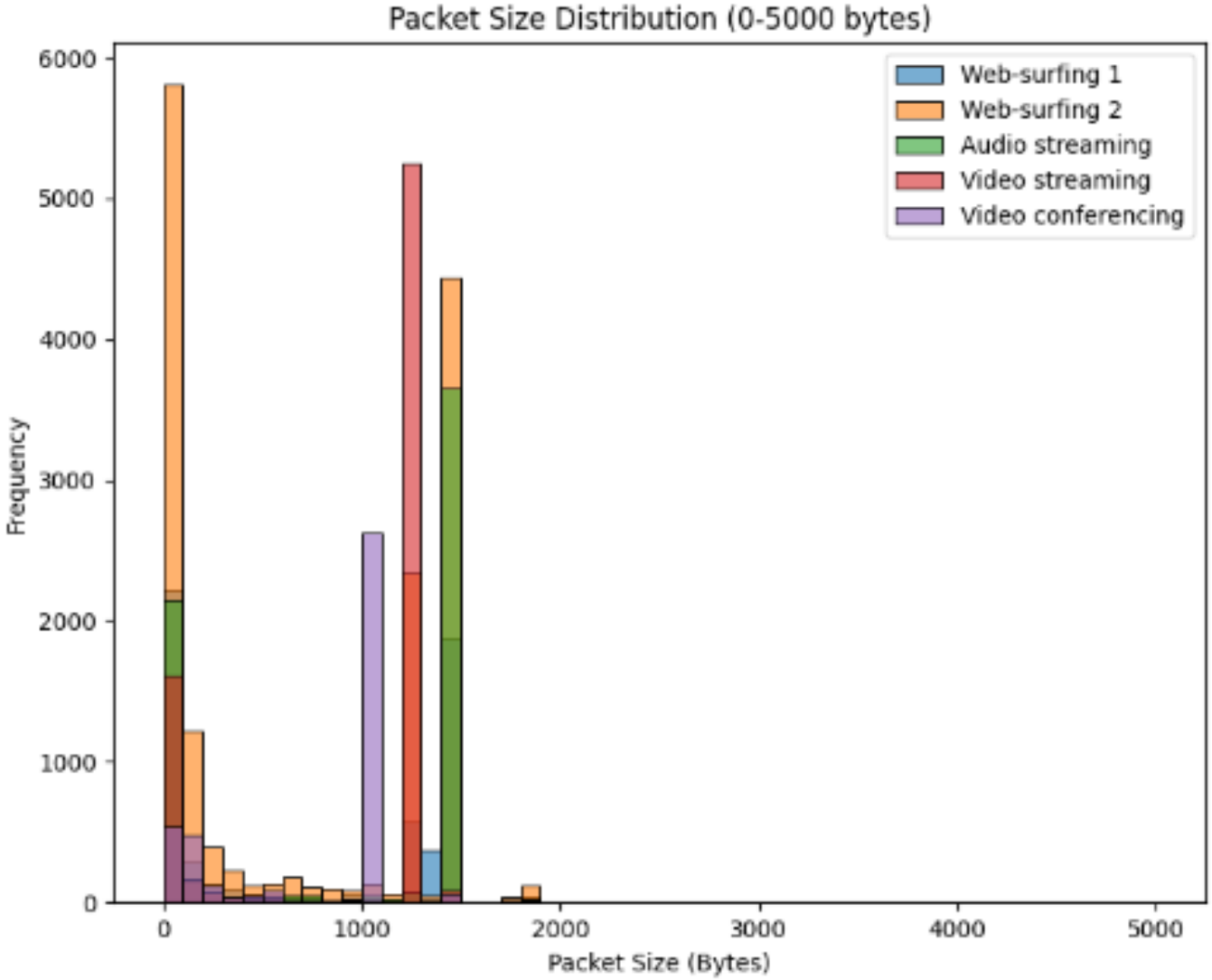
The **key takeaway** is that network behavior patterns can be leveraged for application detection, even when traffic is encrypted. Future research may focus on **machine learning-based traffic classification** and **privacy-enhancing countermeasures.**

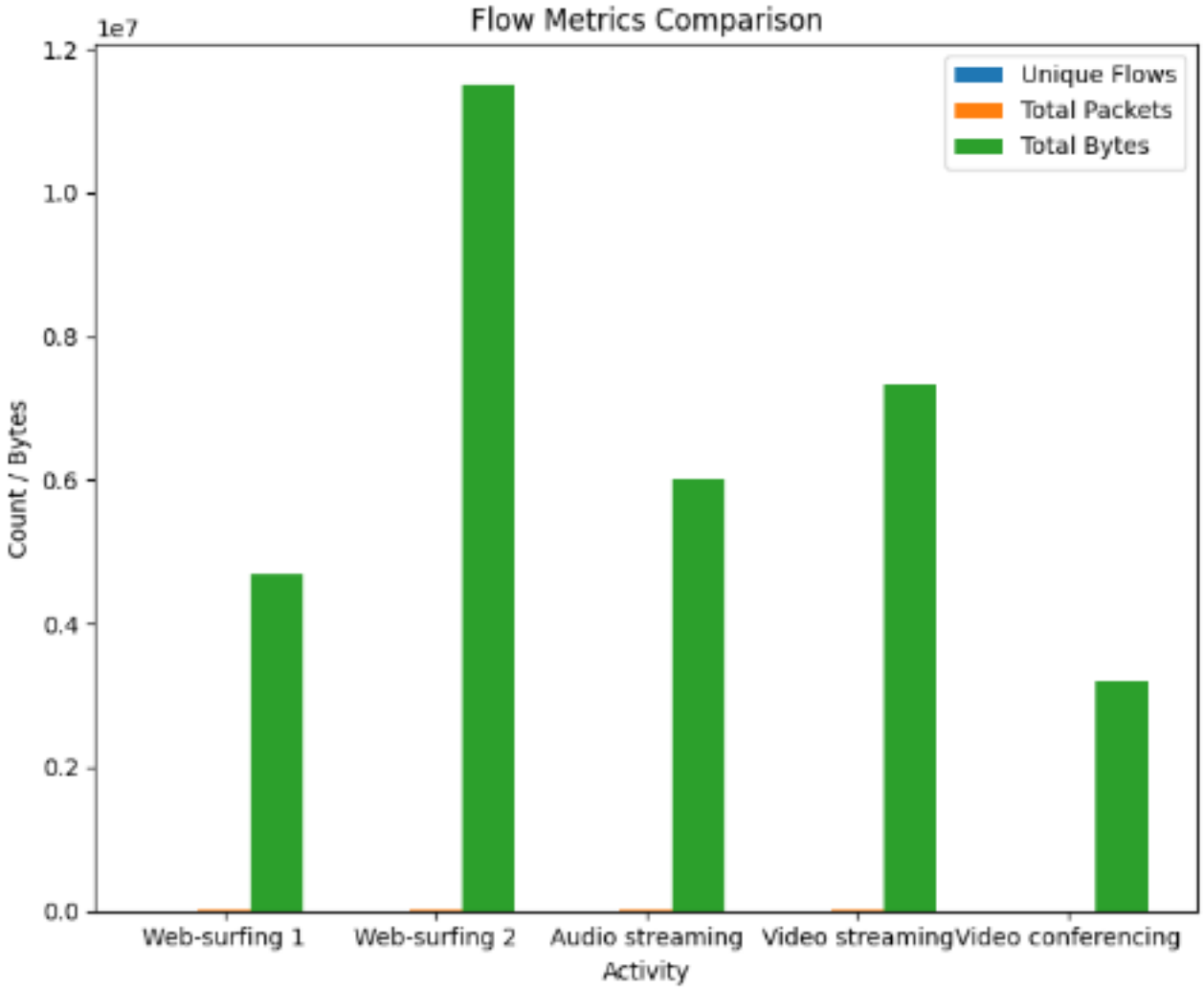
Part 1: Attack with Packet Size, Time Stamp, and 4-Tuple Hash (IP + Port Info)
Represents the attack where the attacker knows each packet's size, time stamp,
and hash of the 4-tuple (source IP, dest IP, source port, dest port).

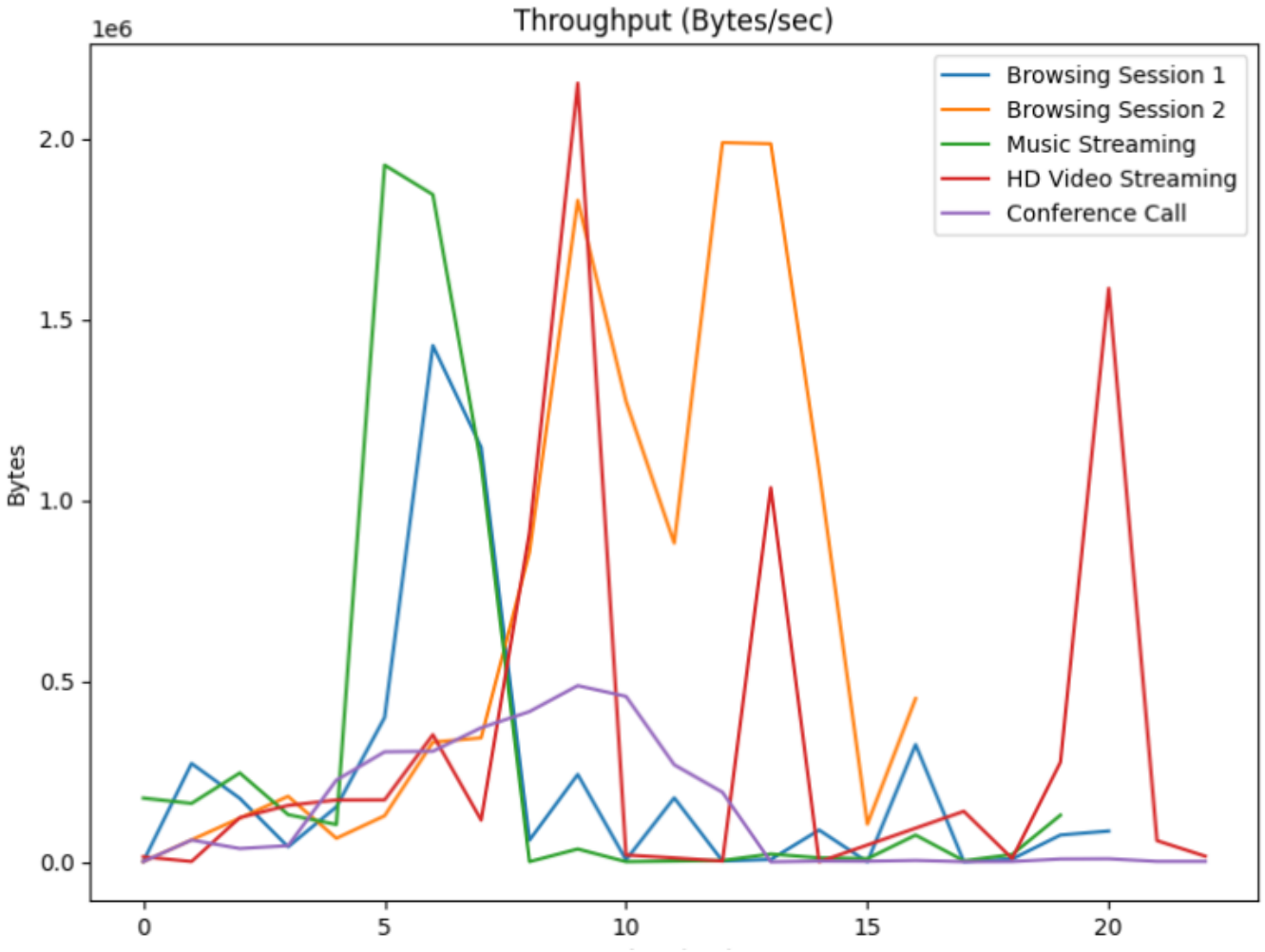
Relevant Graphs:
Flow Metrics Comparison
This graph helps analyze unique flows, total packets, and total bytes per activity.
Since the attacker has access to IP addresses and ports,
they can distinguish different flows more precisely and classify specific applications.
Flow Volume Comparison (Not shown here but similar to Flow Metrics)
If included, this would show the total traffic volume per application,
which is useful when IP and port metadata are available.

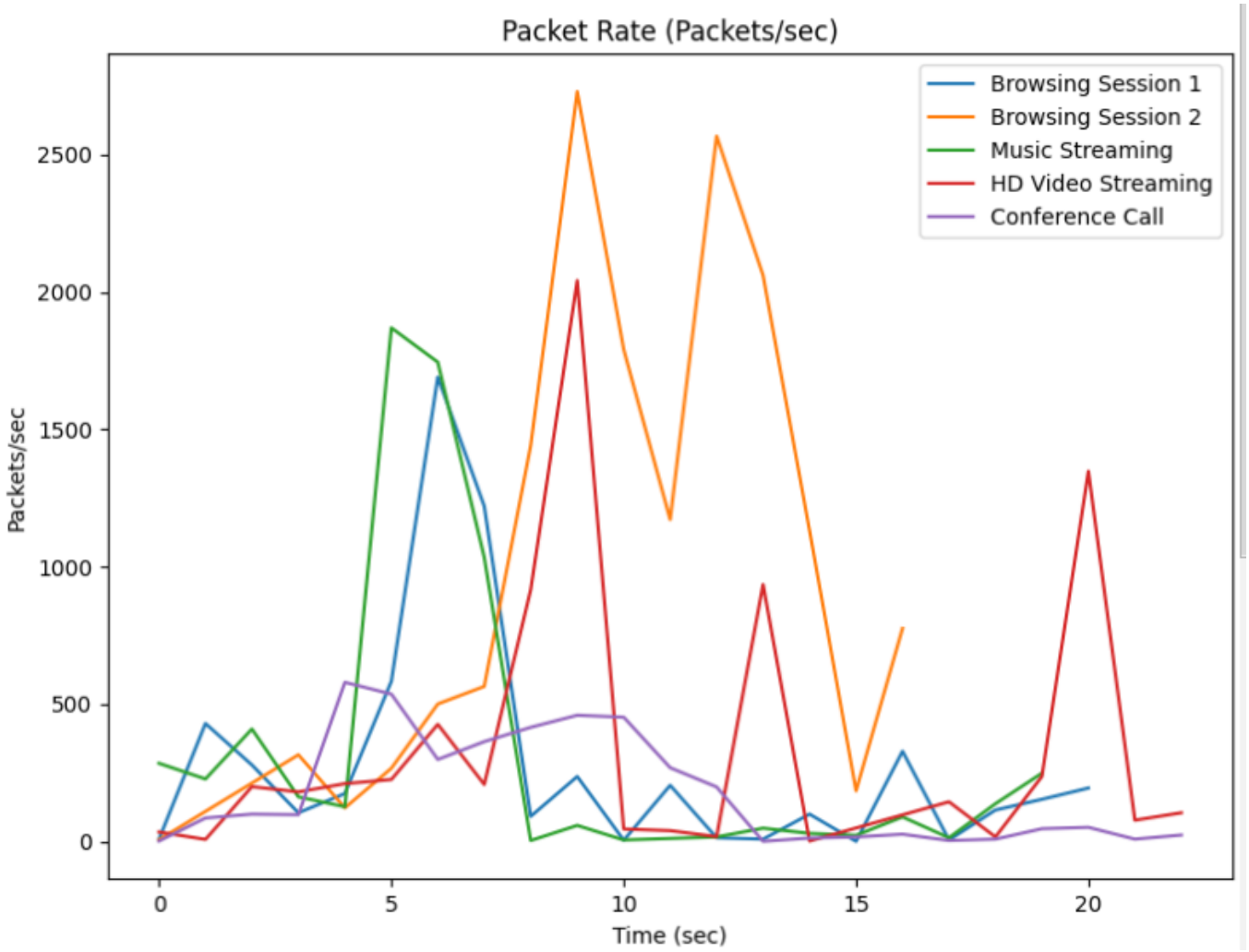
Part 2: Attack with Only Packet Size & Time Stamp
Represents the attack where the attacker knows only
the size and time stamp for each packet.

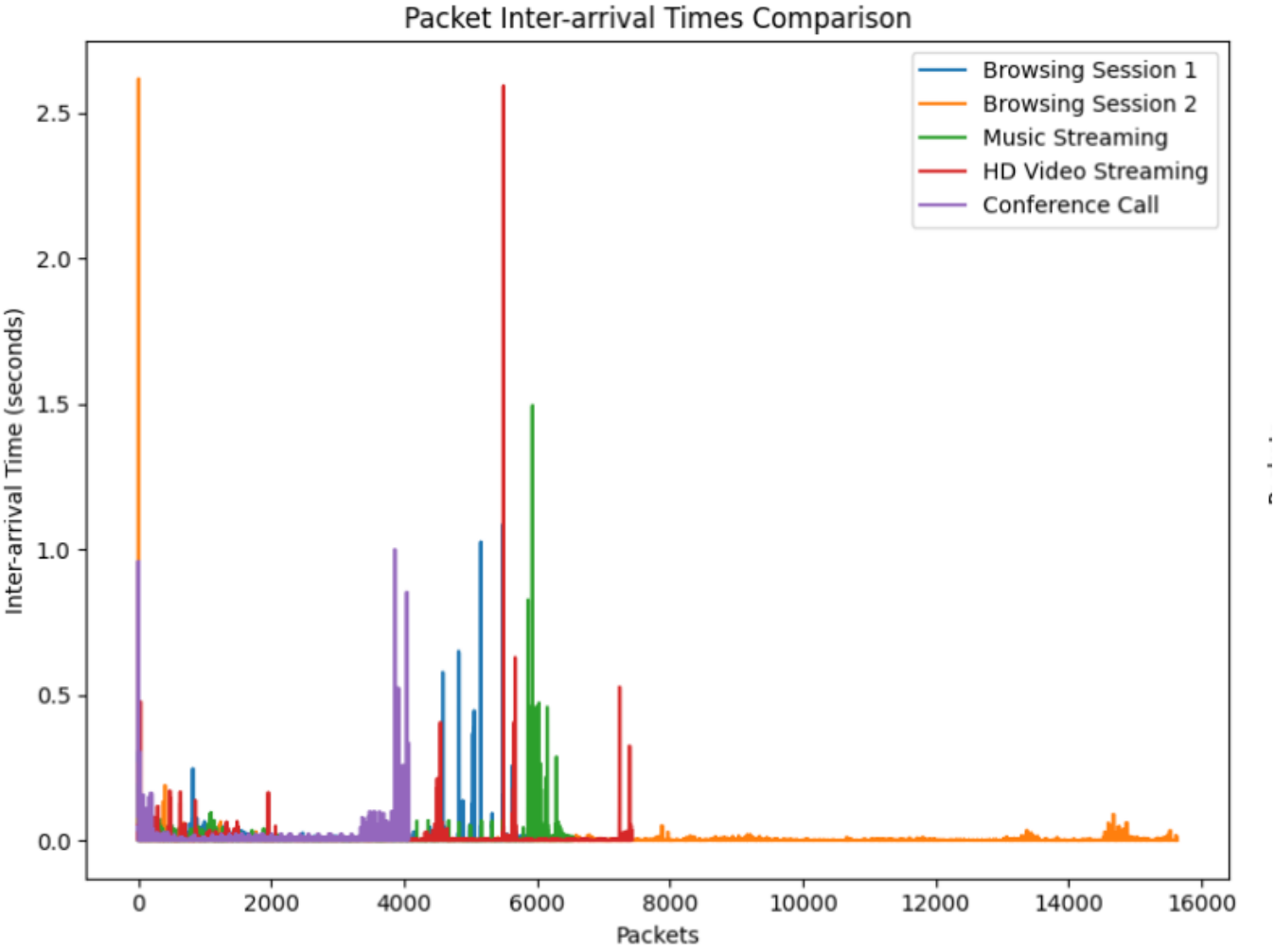
Relevant Graphs:
Packet Size Distribution
This graph shows how packet sizes vary without needing IP or port metadata.
An attacker could still infer different application types
based on packet size distributions.
Inter-Arrival Time Distribution
This graph shows the frequency of packets arriving within short time intervals.
Even without knowing source/destination IPs or ports, an attacker can
observe timing patterns to classify activities like browsing vs. streaming vs. conferencing.

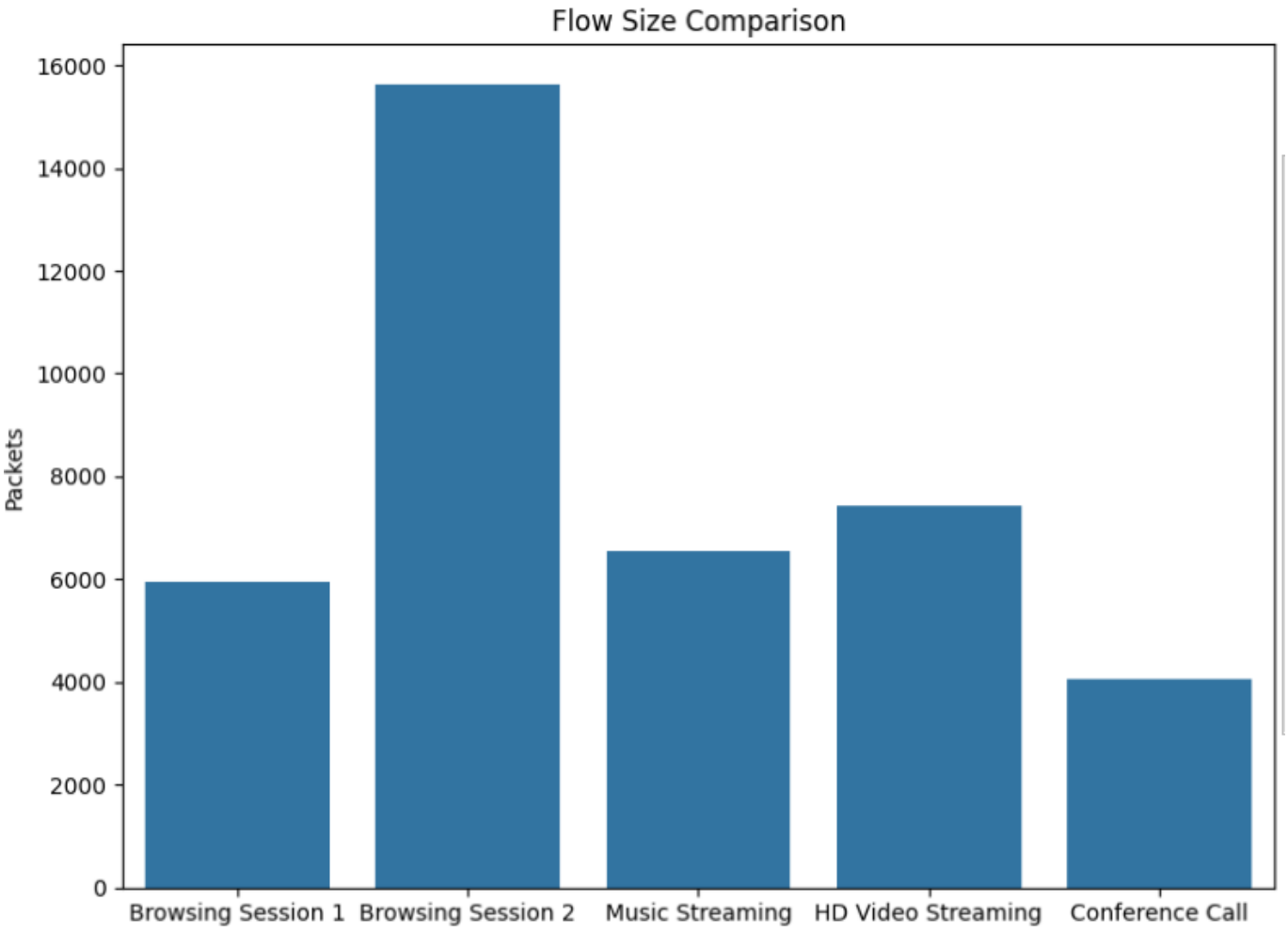

Packet Size Distribution (0-5000 bytes)
Web-surfing 1
Web-surfing 2
Audio streaming
Video streaming
Video conferencing
Frequency
Packet Size (Bytes)
0
1000
2000
3000
4000
5000
6000

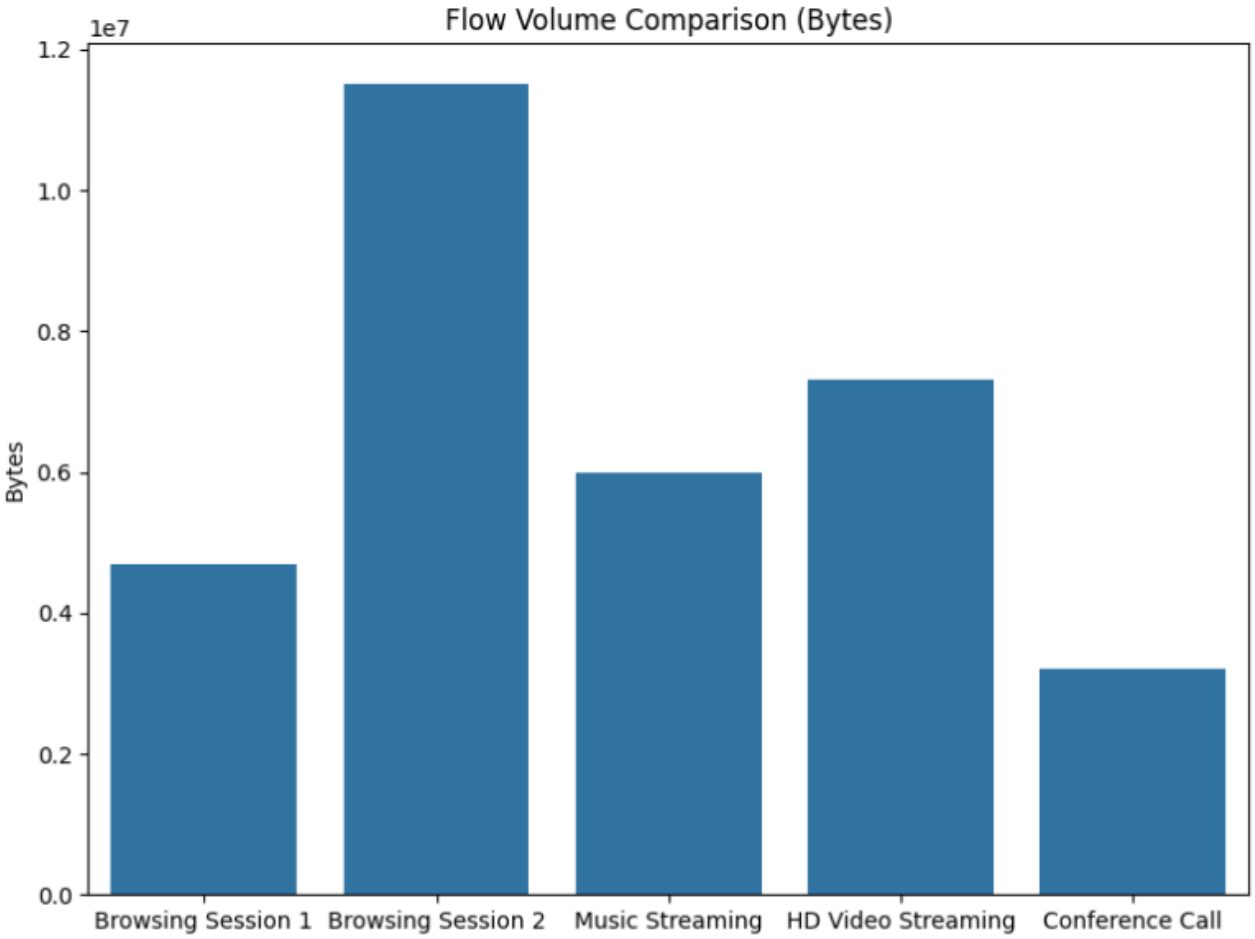

Flow Metrics Comparison
1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Count / Bytes
Unique Flows
Total Packets
Total Bytes
Web-surfing 1
Web-surfing 2
Audio streaming
Video streaming
Video conferencing
Activity


Throughput (Bytes/sec)
1e6
Bytes
2.0
1.5
1.0
0.5
0.0
0
5
10
15
20
Browsing Session 1
Browsing Session 2
Music Streaming
HD Video Streaming
Conference Call


Packet Rate (Packets/sec)
Browsing Session 1
Browsing Session 2
Music Streaming
HD Video Streaming
Conference Call
Packets/sec
2500
2000
1500
1000
500
0
0
5
10
15
20
Time (sec)


Packet Inter-arrival Times Comparison
Browsing Session 1
Browsing Session 2
Music Streaming
HD Video Streaming
Conference Call
Inter-arrival Time (seconds)
0.0
0.5
1.0
1.5
2.0
2.5
0
2000
4000
6000
8000
10000
12000
14000
16000
Packets


Flow Size Comparison
Packets
16000
14000
12000
10000
8000
6000
4000
2000
0
Browsing Session 1
Browsing Session 2
Music Streaming
HD Video Streaming
Conference Call


Flow Volume Comparison (Bytes)
1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Bytes
Browsing Session 1
Browsing Session 2
Music Streaming
HD Video Streaming
Conference Call