מטלת סיום

part 1 - questions

1. A user reports that their file transfer is slow, and you need to analyze the transport layer to identify the potential reasons. What factors could contribute to the slow transfer, and how would you troubleshoot it?

so there are few possible reasons for slow transfer, from the transport layer the main ones are:

- **congestion control**
- **packet loss**
- **RTT**
- **firewall**

congestion control is holding the speed rate of which the packet is sent. Perhaps the window size is too small and not efficient. To fix it I will first check with wireshark if that indeed is the case. Then, I will adjust it in the OS settings.

Packet loss can slow down the speed significantly because of the retransmissions and congestion control. If it seems like there are a lot of retransmissions and logs of packet loss and its systematic and not one-timers it may indicate a problem in lower layers such as the physical - maybe change the cable.

RTT if the value of the rtt is high it may indicate a bottlenecks stations in the packet route, it worth to check with the traceroute tool to find if such hop exists.

fireWall as well as other intermediate tools might slow down the speed, For obvious reasons such as slow computing time for virus scanning ect. fix it by removing them and installing some better tools, or just modify its setting.

2. Analyze the effects of TCP's flow control mechanism on data transmission. How would it impact performance when the sender has significantly higher processing power than the receiver?

The whole point of having the flow control mechanism is to prevent the fast sender from overwhelming the slow receiver, I will detail the impacts here:

- Limited transfer speed
- High RTT
- TCP window size will shrink

Limited transfer speed by the value of the receiver. Even if the sender has abundant bandwidth and processing power, it must wait for the receiver to free up buffer space.

High RTT or latency, due to the multiple consistent ack - response waits.

TCP window size will shrink If the receiver's buffer fills up too quickly, the window size will shrunk dynamically

3. Analyze the role of routing in a network where multiple paths exist between the source and destination. How does the path choice affect network performance, and what factors should be considered in routing decisions?

Routing is basically how the network decides which way to send data from one place to another. When there are multiple ways to get from the source to the destination, the network has to pick the most efficient one.

There are 3 main effects for this choice.

1. **latency/RTT** - shorter or less crowded paths mean data gets there faster
2. **Throughput** - more bandwidth equals more data
3. **Congestion** - If too many packets go through the same path, things slow down.

the factors for choosing the best route:

Shortest Route: Less distance = faster delivery.
Least Busy Route: Avoids slowdowns from too much traffic.
Reliable Connection: Some paths are more stable, so they won't randomly drop packets.
Type of Data: Live streaming needs a fast, steady path, but emails can take a slower one.

4. How does MPTCP improve network performance?

regular TCP uses only one path to send data, but MPTCP can use multiple paths at the same time. meaning - faster speed and less congestion because it balances traffic across different paths, so no single route gets too crowded.

5. You are monitoring network traffic and notice high packet loss between two routers. Analyze the potential causes for packet loss at the Network and Transport Layers and recommend steps to resolve the issue.

Network layer:

- **Congestion:** Too many packets, not enough bandwidth—some get dropped.
- **How to fix?** Check router logs and traffic graphs for congestion.
- **Faulty Hardware:** A bad router, damaged cables, or overheating equipment.
- **How to fix?** Upgrade network hardware if it's old or overheating.
- **Routing Problems:** Loops, bad routing tables, or misconfigurations.
- **How to fix?** Verify routing tables and make sure there are no loops.
- **Interference (Wireless Networks):** Weak Wi-Fi signals, too much noise, or bad signal quality.
- **How to fix?** If using Wi-Fi, reduce interference by switching channels or using a wired connection.

Transport layer:

- **High Latency:** Delays cause TCP timeouts, forcing packets to be resent.
- **How to fix?** Use QoS (Quality of Service) to prioritize important traffic.
- **Buffer Overflows:** If a device's buffer fills up too fast, it starts dropping packets.
- **How to fix?** Increase buffer sizes if devices are struggling to process packets fast enough.
- **TCP Window Size Too Small:** The sender is limited in how much data it can send at once.
- **How to fix?** Adjust TCP window scaling to allow more data per transmission.

part 2 - paper summarize

FlowPic

• What is the main contribution of the paper?

The main contribution of this article is to show a different creative and most important - efficient (by resources and run-time) solution for the classification of internet traffic both for categorizing traffic types and for identifying specific applications, based **only** on time and size related information.

• What traffic features does the paper use, and which are novel?

The solution uses the packet arrival time and the packet size data. What's novel about it is that it turns this data into a picture and classifies it by using Convolutional Neural Networks or CNN.

• What are the main results (you may copy the figures from the paper), and what are the insights from their results?

Problem	FlowPic Acc. (%)	Best Previous Result	Remark
Non-VPN Traffic Categorization	85.0	84.0 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
VPN Traffic Categorization	98.4	98.6 % Acc., Wang *et al.* [7]	[7] Classify raw packets data. Not including browsing category
Tor Traffic Categorization	67.8	84.3 % Pr., Gil *et al.* [15]	Different categories. [15] used unbalanced dataset
Non-VPN Class vs. All	97.0 (Average)	No previous results	
VPN Class vs. All	99.7 (Average)	No previous results	
Tor Class vs. All	85.7 (Average)	No previous results	
Encryption Techniques	88.4	99. % Acc., Wang *et al.* [7]	[7] Classify raw packets data, not including Tor category
Applications Identification	99.7	93.9 % Acc., Yamansavascilar *et al.* [10]	Different classes

the figure above shows the dry data of the solution, for example - **Non-VPN traffic categorization: 85.0%** (better than previous works as the paper mentioned)

There is a few note regarding the results that are not expressed in the table above:

- The Application identification: 99.7% is much Higher than the old tools they used.
- the reason why Tor traffic categorization: 67.8% is so low is due to Tor's heavy encryption
- As mentioned in the paper the model successfully classifies **new applications** even if they were **not part of the training data,** which is huge in terms of AI.
- because the model uses images instead of data, it allows it to excel in encrypted data classification, because it doesn't mind what is inside the data.

All the other results are in the table above. There is no point in copying it here if there is no special achievement worth noting outside the table (there is nothing to add).

Early Traffic

• What is the main contribution of the paper?

The primary contribution of the article lies in the development of a novel algorithm, termed hRFTC (Hybrid Random Forest Traffic Classifier), designed to enhance the early classification of encrypted traffic within communication networks. This innovation specifically addresses the challenges introduced by Encrypted ClientHello (ECH) in the TLS 1.3 protocol. The hRFTC algorithm integrates packet-based features and flow-based features in a hybrid approach, facilitating precise identification of traffic types—such as video, audio, or web browsing—and their associated Quality of Service (QoS) requirements, even when critical identifiers like the Server Name Indication (SNI) are obscured by ECH encryption.

• What traffic features does the paper use, and which are novel?

The hybrid synthesis of packet-based and flow-based features, adapted to advanced encryption protocols like ECH and QUIC, constitutes the core innovation. This method ensures effective classification under constrained conditions.

Packet-based Features(such as the ClientHello):
TLS Version: Specifies the protocol version (e.g., TLS 1.3).
Cipher Suites: Enumerates the encryption options offered by the client.
Length and Types of Extensions: Captures metadata about additional functionalities, such as "Key Share" or "Supported Versions."
Although these features are not novel in isolation, their application in the context of ECH represents a significant advancement. With SNI encrypted, hRFTC creatively exploits residual unencrypted data to infer traffic characteristics. Furthermore, the article extends this methodology also to the QUIC protocol (underpinning HTTP/3).

Flow-based Features:
Packet Sizes.
Inter-Packet Times.
Flow Statistics: Includes mean, variance, minimum, and maximum values of packet sizes and inter-arrival times.
Initial Packet Patterns: the sequence of initial packets prior to the onset of application data transmission.
The distinctive aspect here is the dynamic flow analysis strategy. Rather than relying on a predetermined packet count, hRFTC analyzes the initial packet sequence up to the start of application data, enabling rapid classification.

• What are the main results (you may copy the figures from the paper), and what are the insights from their results?

TABLE 11. Full dataset per class F-score for different classifiers.

Class	F-score [%]						
	Hybrid Classifiers			Flow-based Classifier	Packet-based Classifiers		
	hRFTC [proposed]	**UW** [35]	**hC4.5** [34]	**CESNET** [63]	**RB-RF** [24]	**MATEC** [33]	**BGRUA** [32]
BA-AppleMusic	**92.1**	89.5	80.2	89.2	25.5	13.1	14.5
BA-SoundCloud	**99.6**	98.9	97.8	98.7	84.4	81.8	82.0
BA-Spotify	**93.6**	90.8	89.0	88.5	16.3	0.0	3.6
BA-VkMusic	**95.7**	89.7	88.5	91.8	2.6	2.1	3.2
BA-YandexMusic	**98.5**	93.2	93.7	92.5	1.8	0.2	0.1
LV-Facebook	**100.0**	99.7	99.8	99.8	**100.0**	**100.0**	**100.0**
LV-YouTube	**100.0**	**100.0**	99.9	**100.0**	**100.0**	99.0	98.4
SBV-Instagram	**89.7**	74.7	76.5	78.8	10.0	6.3	6.4
SBV-TikTok	**93.3**	81.8	81.8	76.3	38.3	34.3	34.5
SBV-VkClips	**95.7**	94.0	91.3	92.4	53.2	37.7	46.0
SBV-YouTube	**98.2**	96.6	94.7	96.4	1.1	0.2	0.2
BV-Facebook	**87.7**	78.2	79.7	77.6	5.6	3.2	3.8
BV-Kinopoisk	**94.1**	84.1	85.8	89.8	5.4	4.0	4.1
BV-Netflix	**98.5**	97.2	95.2	93.7	50.7	52.3	56.1
BV-PrimeVideo	**91.3**	86.7	84.1	84.7	32.5	24.7	26.8
BV-Vimeo	**94.8**	90.5	90.2	81.4	72.0	19.5	68.6
BV-VkVideo	**88.6**	80.5	80.4	79.7	10.5	0.0	0.1
BV-YouTube	**85.9**	84.3	77.0	78.5	22.3	19.6	20.2
Web (known)	**99.7**	99.5	99.4	99.4	98.0	98.0	98.0
Macro-F-score (average)	**94.6**	89.9	88.7	88.9	38.4	31.4	35.1

LV is Live Video, (S)BV is (Short) Buffered Video, and BA is Buffered Audio.

Main Results

- Classification Accuracy: hRFTC attains a 94.6% Macro F-score across 19 traffic categories, outperforming packet-based algorithms (e.g., RB-RF at 38.4%) and other hybrid models (e.g., hC4.5 at 89%).
- Feature Importance: Using the Gini-Impurity metric, the length of the Cipher Suites list and Downlink packet size statistics emerge as the most influential features, underscoring the synergy between packet and flow data.
- Geographic Variability: Accuracy declines to 38.4% when applied to data from an untrained region (e.g., Germany), contrasting with strong performance on locally trained datasets.

Analyzing HTTPS

• What is the main contribution of the paper?
The primary contribution of this paper lies in its ability to identify the triplet <operating system, browser, application> of a user by analyzing encrypted HTTPS traffic in a passive manner—that is, without any interference with the user's device activity—achieving an impressive accuracy rate of 96.06%. This marks a significant advancement over prior research, which typically concentrated on extracting information from unencrypted traffic or identifying isolated components (such as specific applications), rather than the complete triplet.

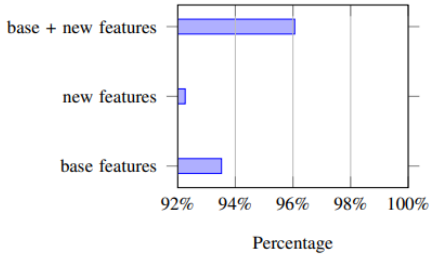
• What traffic features does the paper use, and which are novel?

Refer to the entry:

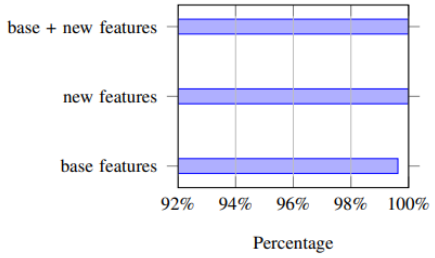
"TABLE I: The two sets of features used in this paper. The base features are features that are used in many traffic classification methods. The new features are proposed in this paper."

In addition to these, the paper introduces a central innovative concept: leveraging the bursty behavior of browsers as a key identifier. The study reveals that different browsers and operating systems generate distinct traffic patterns during "bursts"—periods of non-continuous data transmission. This approach represents a novel contribution, unseen in previous studies targeting this triplet, enabling more precise differentiation among various combinations of operating systems, browsers, and applications.

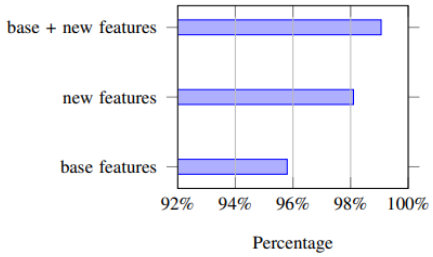
• What are the main results (you may copy the figures from the paper), and what are the insights from their results?
The key results demonstrate an enhancement in accuracy, reaching 96.06% when combining both basic and newly proposed features. (See attached Figure 2.)



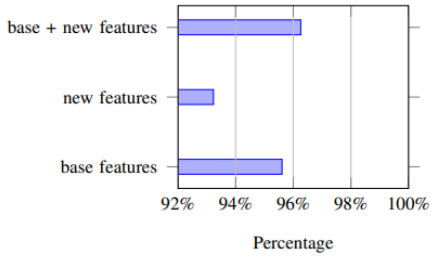
(a) Tuple Accuracy Results



(b) OS Accuracy Results



(c) Browser Accuracy Results



(d) Application Accuracy Results

the insights derived from these findings:

- Vulnerability of HTTPS Traffic: The results underscore that encrypting content via HTTPS does not fully safeguard user privacy. Traffic patterns, which remain accessible to a passive observer, expose sensitive details—such as the operating system and browser—potentially enabling attackers to craft targeted exploits.
- Contribution of New Features: The accuracy improvement from 93.51% to 96.06%, driven by the inclusion of new features like burstiness, highlights the value of analyzing network behavior at a broader level. This serves as a critical lesson in computer communications: valuable insights can be gleaned not just from individual packets, but from overarching traffic patterns.
- Contribution of New Features: The accuracy improvement from 93.51% to 96.06%, driven by the inclusion of new features like burstiness, highlights the value of analyzing network behavior at a broader level, And the important insight that valuable insights can be gleaned not just from individual packets, but from overarching traffic patterns.
- Future Challenges: These findings prompt critical questions about the future of network security—namely, how traffic behavior might be modified to thwart such analysis, while still maintaining performance efficiency.

The total accuracy of the classification is 96.06%.

<u>part 3 - code explanation</u>

github: https://github.com/Aviv-Avichail/net_project

<u>apps/sites that were recorded:</u>

- chrome,
- google meets,
- youtube,
- spotify,
- edge

characteristics that were measured and are being compared:

- Inter-arrival packets time
- upstream-downstream distribution,
- ip version(v6 or v4),
- packet sizes,
- ip addresses destinations,
- destination ports

use of every characteristic:

Inter-arrival Packets Time

- **Use:** Measures the time between consecutive packet arrivals. Used for traffic pattern analysis, anomaly detection (e.g., DDoS attacks), and QoS evaluation.

Upstream-Downstream Distribution

- **Use:** Analyzes the proportion of data sent upstream (from client to server) versus downstream (from server to client). Used to distinguish different types of network applications (e.g., video streaming vs. web browsing) and detect asymmetric traffic flows.

IP Version (v6 or v4)

- **Use:** Identifies whether IPv4 or IPv6 is used in communication. Used for protocol transition analysis, compliance checks, and understanding network adoption trends.

Packet Sizes

- **Use:** Examines the distribution of packet sizes in network traffic. Used for application classification, performance optimization, and detecting anomalies like fragmentation attacks.

IP Addresses Destinations

- **Use:** Identifies the target endpoints of network traffic. Used for geolocation analysis, security monitoring (e.g., detecting malicious hosts), and load balancing.

Destination Ports

- **Use:** Determines the services and applications being accessed (e.g., HTTP on port 80, HTTPS on port 443). Used for network security (firewall rules), traffic classification, and protocol analysis.

Scenario 1: Attacker with Complete Information (Packet Size, Timestamps, and Flow ID)

Traffic Characteristics by Application Type

Media Streaming (YouTube/Spotify)

- **Distinctive Traffic Patterns**: Significantly sized packets (ranging from 100KB-1MB) with high continuity
- **Buffering Behavior**: Identifiable initial data bursts for buffer filling, followed by more stable transmission
- **Video vs. Audio Differences**: Video traffic exhibits I-frame patterns (significantly larger sizes) at regular time intervals, while audio shows more uniformity in packet sizes
- **Asymmetric Ratio**: Substantially higher volume of incoming data compared to outgoing, often exceeding a 10:1 ratio

Web Browsers (Chrome/Edge)

- **Packet Size Distribution**: Broader distribution of packet sizes, ranging from very small packets (a few bytes) to medium-sized packets (tens of KB)
- **Multiple Parallel Streams**: Traffic from diverse sources (various content servers) corresponding to varied Flow IDs
- **Request-Response Patterns**: Identifiable patterns of small requests followed by larger responses
- **Random Data Bursts**: Less predictable patterns dependent on website content

Multi-participant Video Calls (Zoom)

- **Balanced Bidirectional Traffic**: More balanced ratio between incoming and outgoing traffic
- **Adaptive Packet Size**: Dynamic changes in packet sizes according to the number of participants and activity
- **Time Sensitivity**: More consistent time intervals between packets, indicating real-time communication requirements

- **Precise Mapping**: Ability to correlate packets with specific streams enables isolation of application traffic
- **Server Identification**: Possible correlation between server IP addresses and known services (e.g., Google's IP ranges versus Microsoft's)
- **Protocol-Based Analysis**: Identification of protocols used by specific applications
- **Rich Statistical Analysis**: Capability to build precise statistical models for each stream separately

we can mitigate the attack by:

1. Using VPN and Traffic Obfuscation

<u>VPN:</u>
Hides the user's original IP address and routes all traffic through an external server. Severs the link between the traffic and the original IP addresses/ports, complicating flow identification.

<u>Anonymous Networks (e.g., Tor):</u>
Mixes traffic from multiple users, preventing packet association with specific users/apps.

2. Enhancing Encryption

<u>Advanced Encryption (SSL/TLS):</u>
Prevents attackers from reading packet content, even if they collect metadata.

<u>Anonymous Protocols (e.g., QUIC):</u>
Obscures flow identification by encrypting headers and data.

<u>Obfuscation Tunnels:</u>
Uses protocols to hide metadata, e.g., IPs and ports.

3. Modifying Packet Sizes

<u>Padding:</u>
Adds redundant bytes to standardize packet sizes, e.g., 1,500 bytes.

<u>Fragmentation:</u>
Splits data into inconsistently sized packets to disrupt statistical patterns.

4. Adding Noise

Dummy Packets:
Sends empty or meaningless packets at random intervals to mask real traffic patterns.

Traffic Noise:
Runs background apps to generate decoy traffic that distracts attackers.

5. Altering Packet Arrival Times

Random Delays:
Introduces artificial latency, e.g., random 10–200 ms delays, to break fixed timing patterns.

Replay Scheduling:
Transmits packets in a non-linear order while maintaining data integrity on the receiver's side.

Scenario 2: Attacker with Limited Information (Only Packet Size and Timestamps)

In this case, the attacker does not have flow IDs, source, or destination information. Instead, they can only analyze packet sizes and timestamps. While this significantly reduces their ability to link traffic to specific services, some inferences can still be made.

How the attacker can still gather information:

- **Media Streaming (YouTube/Spotify)**
  - Large packets at regular intervals may still suggest video or audio streaming.
  - The initial buffering burst followed by steady traffic remains visible.
- **Web Browsers (Chrome/Edge)**
  - Small bursts of packets followed by larger responses suggest a request-response pattern typical of web browsing.
  - Harder to link specific websites but still possible to detect general browsing activity.
- **Multi-participant Video Calls (Zoom)**
  - Consistent packet intervals indicate real-time communication.
  - More evenly balanced incoming and outgoing traffic.

Although the attacker lacks direct IP data, statistical traffic analysis could still help them distinguish between general categories of applications.

How to mitigate this attack:

1. **Standardizing Traffic Patterns**

    - **Traffic Shaping:** Adjusts packet flow to make all traffic appear uniform.
    - **Constant Bitrate Streaming:** Makes all outgoing traffic look the same regardless of content.
2. **Packet Size Obfuscation**

    - **Padding and Fragmentation:** Prevents easy classification by modifying packet sizes.
3. **Dummy Traffic and Background Noise**

    - **Injecting random traffic** to create misleading patterns.
    - **Running background applications** to make real traffic harder to isolate.
4. **Time-based Countermeasures**

    - **Adding jitter (random delays)** to prevent timing-based fingerprinting.
    - **Batching transmissions** to obscure real-time patterns.

By applying these techniques, users can significantly reduce the effectiveness of traffic analysis, even when an attacker is monitoring encrypted or anonymized traffic.

To sum up section 4, the main difference between these scenarios is the attacker's ability to track specific connections. In Scenario 1, the attacker has access to flow IDs, allowing them to differentiate between separate connections and correlate them with known services. This means they can track **distinct application usage**, identify **specific websites**, and even infer **user behavior** over time. In contrast, in Scenario 2, the attacker only sees packet sizes and timestamps, making it much harder to distinguish between different applications. While they can still infer general activity types (e.g., streaming vs. web browsing), they **cannot reliably link traffic to specific services**. Without flow information, distinguishing between multiple simultaneous activities also becomes challenging, reducing the precision of traffic analysis.