Contents lists available at IdNNS

Indonesian Journal of Cyber-AI and Security Intelligence

Volume 1 ; Issue 1

journal homepage: https://journal.idnns.org/index.php/ijcasi

Revisiting OSI Layer 4 in Traditional DPI: A Systematic Thematic Review of TCP Header Fields for Anomaly Detection

Shafana Muhammed Shareef[a], Adamu Abubakar Ibrahim[b]

[a] Kulliyyah of Information and Communication Technology, PInternational Islamic University Malaysia, Kuala Lumpur, Malaysia
[b] Department of Information and Communication Technology, South Eastern University of Sri Lanka, , Oluvil, Sri Lanka

ARTICLE INFO

Keywords:
Anomaly Detection
ATLAS.ti
Checksum
Control Flags
Deep Packet Inspection
Network Security
Sequence Number
TCP Header
TCP Header Fields
Transport-Layer Analysis
Systematic Review
Thematic Synthesis

ABSTRACT

TCP remains a common transport substrate in operational networks, yet traditional deep packet inspection (DPI) anomaly detection studies report TCP header usage in a narrow and unevenly documented way. Transport-layer header cues are often treated as deterministic signals for steering inspection, but the literature is fragmented in how fields are selected and operationalized across parsing, flow construction, feature extraction, and rule or policy gating. To consolidate evidence, this study conducts a thematic review of TCP header-field usage as reported within traditional DPI anomaly detection studies over a six-year window (2020–2025), using a code-to-document workflow in ATLAS.ti that codes each eligible study by field and functional role. Findings are structured using a Realist Context–Feature–Mechanism–Outcome (CFMO) lens grounded in CMO logic and the view that mechanisms are resources plus reasoning, treating TCP fields as resources (features) that enable DPI mechanisms, consistent with CIMO-style extensions. Publication activity is modest with a peak in 2021, a dip in 2023, a rebound in 2024, and a marked decline in 2025. Destination and source ports are the most consistently reported cues, supporting five-tuple context, service/protocol demultiplexing, and rule or policy selection, while control flags provide state semantics for handshake validation, teardown checks, and misuse screening. Sequence and acknowledgment numbers appear less often and support continuity and manipulation contexts, and data offset anchors header-length validation and payload alignment. Window size is used selectively as a recorded feature, TCP options are rare and mainly reflect MSS evidence in SYN packets, and checksum is sparsely reported and typically appears in niche contexts rather than as a primary detector; reserved bits, the urgent pointer, and header padding are absent as explicit themes. The review contributes a concise TCP field-to-function taxonomy aligned to CFMO, highlights reporting gaps that constrain reproducibility, and proposes an agenda centered on explicit extraction and validation pipelines, baselines for flag and handshake patterns, option-aware parsing and window-scale considerations, transparent checksum interpretation under capture artifacts, and robustness testing under NAT, proxies, asymmetric paths, and middlebox remapping.

1. Introduction

Traditional deep packet inspection looks inside packet payloads and reads header metadata to find behavior that breaks stated rules and protocol expectations (Góngora-Blandón and Vargas-Lombardo, 2012). It rebuilds sessions, cleans odd encodings, and pulls simple fields as traffic moves through the stack, which gives a clear view of each flow for checking (**?**). The system then compares these parts with fixed rules and signature sets, so byte strings, tokens, or grammar states create exact matches that support clear actions. Protocol decoders check order and format with strict state machines, therefore malformed fields or strange sequences raise alerts that analysts can explain (**?**). Deployed at gateways and middleboxes, such tools act inline, so they can block content, rewrite parts, or log accountable evidence for later review. Limits on speed and memory shape parser depth and state retention, so implementations focus on evidence that is quick to extract and easy to read. Even though encryption hides much of the content, message sizes, timing patterns, and visible header values still guide rule checks in practice. Attackers try to hide with fragmentation, ambiguous encodings, and protocol mimicry, so robust DPI uses normalization and tight conformance checks to stay dependable (Handley et al., 2001).

1.1. Traditional DPI and TCP Header Fields

Traditional deep packet inspection checks TCP traffic with fixed, rule based parsing, and it uses the header to guide what to read in the payload (Góngora-Blandón and Vargas-Lombardo, 2012; **?**). Source and destination ports support service mapping and demultiplexing, so a flow can be tied

ORCID(s):

to an application and to the right policy. Flags such as SYN, ACK, FIN, and RST drive connection setup and teardown, therefore odd or impossible flag mixes help detectors find errors and abuse (**?**). Sequence and acknowledgment numbers enable reliability and ordering, so gaps, overlaps, and unexpected jumps become visible during reassembly. The data offset tells where the payload starts, which lets the parser skip any options cleanly and check header lengths for sanity. The advertised window, together with window scale, expresses receive capacity, so rate limits and stalled peers can be noticed early. Options such as MSS, SACK, and timestamps act as capability hints and path fit signals, therefore ill formed or unlikely values are easy to flag (Allman et al., 1999). The checksum provides basic integrity for the header and the payload, which lets rules ignore damaged segments and trust clean ones. Less common fields, including the urgent pointer and reserved bits, still matter in practice, because their presence or misuse can mark legacy software or evasion attempts. Across these fields, traditional DPI applies clear, standard driven checks, it normalizes strange cases, and it records simple evidence that analysts can explain.

2. Objective of the Thematic Review

This thematic review examines how traditional, deterministic, rule based DPI uses TCP transport layer header fields for anomaly detection in studies published from 2020 to 2025, and it explicitly excludes work that relies only on machine learning without any traditional technique. The synthesis focuses on two axes: (1) usage trends for each field, including Source Port, Destination Port, Sequence Number, Acknowledgment Number, Data Offset, Flags, Window Size, Checksum, Urgent Pointer, and common Options such as MSS, Window Scale, SACK, and Timestamps, and (2) the inspection intent linked to these fields, for example service mapping and demultiplexing, connection state tracking and policy enforcement, reliability and ordering checks, flow control assessment and capability signaling, integrity verification through the checksum and pseudo header, and evasion resistance through normalization and consistency controls. By mapping these patterns across recent literature, the review consolidates how header and payload co inspection is practiced in TCP aware, traditional DPI and highlights which fields are emphasized, which fields are underused, and which fields are changing in prominence.

In line with these aims, the review is guided by the following question:

Guiding question: What patterns and trends characterize the use of Transport Layer TCP header fields in traditional DPI for anomaly detection between 2020 and 2025?

2.1. Theoretical Framework

2.2. Theoretical Framework

We use (as presented in Figure 1) a Realist Context–Feature–Mechanism–Outcome (CFMO) lens grounded in CMO logic (**?**) and the view that mechanisms are resources plus reasoning (Dalkin et al., 2015); we therefore make the resource explicit as Feature, akin to CIMO-style extensions (Denyer et al., 2008). Features are TCP header fields (Source Port, Destination Port, Control Flags, Sequence Number, Acknowledgment Number, Data Offset, Window Size, Options, and Checksum) treated as resources that enable mechanisms in traditional DPI, namely demultiplexing and parser branching, flow/session keying and correlation, service mapping and rule gating or policy selection, state tracking for handshake validation, teardown checks and misuse screening, ordering continuity and reassembly support, header-length validation and payload-boundary alignment, capability cues via options, and integrity-related handling including forged-packet checks. Context comprises micro-contexts explicitly reported (NAT/ephemeral reuse, proxies/load balancers and port remapping, service multiplexing on common ports, asymmetric paths/routing variance, option clamping or stripping, NIC offload and mirror-tap artifacts, IPv4 versus IPv6 checksum-handling differences, and tool handling/logging defaults) and a macro-context (Doc-Country/Region derived from author affiliations). Outcomes summarize, per Feature→Mechanism linkage, the usage pattern (frequency, bundles, trends by year/country), advantages (benefits), and limitations (constraints/failure modes). Because few studies attribute quantitative performance to individual TCP fields, outcomes are synthesized qualitatively (**?**).

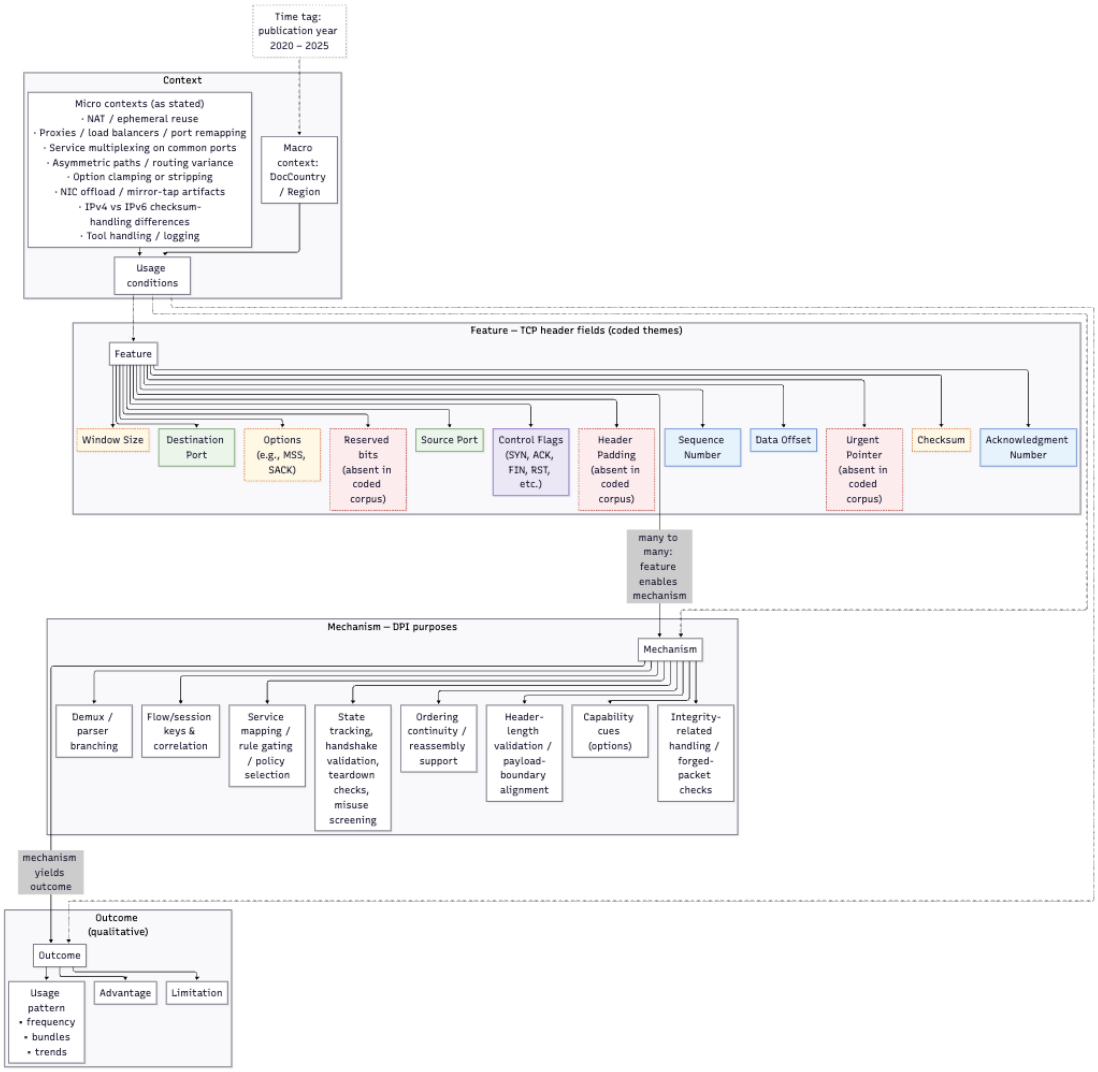

Figure 1: CFMO based Theoretical Framework of this Thematic analysis

3. Rationale for Employing a Thematic Review

While systematic reviews aggregate findings and rate quality in a repeatable way (**?**), our aim is to clarify how TCP header fields are actually used inside traditional, rule based DPI. Because many studies mix header checks with policies or payload signatures, attributing gains to headers alone is unsafe, so we treat such claims with care.

We therefore chart recurring uses of ports, flags, sequence and acknowledgment numbers, data offset, window, checksum, urgent pointer, and common options such as MSS, window scale, SACK, and timestamps, together with their inspection intents, for example demultiplexing, state tracking, ordering checks, flow control assessment, and integrity verification with the pseudo header. We also note constraints that shape per field inspection, including NAT and ephemeral port reuse, asymmetric paths and load balancing, middleboxes that clamp MSS or strip options, and offload features that confuse simple validators. A thematic review offers the breadth to group diverse studies by usage trends and inspection intent, revealing practice focused patterns that a purely quantitative synthesis would miss.

4. Search String

This literature review set out to capture the latest work on how traditional deep packet inspection tools use TCP transport-layer header fields to spot anomalous traffic between 2020 and 2025. Because authors describe DPI in many different ways, the initial search terms were kept broad. To avoid missing relevant work, we first targeted traditional DPI in general, not TCP specifically and refined results with exclusion rules later. At this stage, field-specific words such as “SYN,” “ACK,” “FIN,” “RST,” “sequence number,” “window,” “MSS,” “SACK,” or “timestamps” were deliberately left out to reduce false negatives. Only technically rich papers that showed real DPI operations for example filter

chains, rule matching, protocol decoding, or header-payload co-inspection were included; purely theoretical discussions were screened out.

A uniform Boolean string was applied to every database:

("Deep Packet Inspection" OR "Packet Filtering" OR "Packet Inspection" OR "Traffic Inspection" OR "Payload Inspection" OR "Encryption Inspection") AND "Anomaly Detection".

The search covered peer-reviewed journals and conference proceedings published in English across eight major digital libraries: IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Scopus, ProQuest, Taylor & Francis, and Wiley Online Library.

5. Inclusion and Exclusion Strategy

To keep the review methodical and on topic, we applied clear selection rules to articles. We included only peer reviewed journal or conference papers published in English between 2020 and 2025 that described rule based, traditional DPI inspecting both TCP header fields and packet payloads for anomaly detection in network security contexts. Papers were excluded if they predated 2020, were not in English, came from non scholarly outlets such as blogs or white papers, omitted header plus payload co inspection, focused only on machine learning or flow only statistics, or had been retracted. Selection proceeded in four tiers, title screening removed unrelated topics, abstract checks dropped off theme or shallow work, a quick diagonal read confirmed genuine DPI depth, and a full text assessment verified practical use of TCP header fields together with content parsing. Eligibility at the final tier required explicit handling of fields such as Source Port, Destination Port, Sequence Number, Acknowledgment Number, Data Offset, Flags, Window Size, Checksum with pseudo header rules under IPv4 and IPv6, Urgent Pointer, and common Options including MSS, Window Scale, SACK, and Timestamps.

Studies were retained when they demonstrated concrete DPI tasks, for example filter chains, protocol decoding, port based demultiplexing, signature or regex matching, checksum validation, state machine tracking of SYN, ACK, FIN, and RST, sequence and window analysis for ordering and flow control, option parsing, and fine grained payload inspection at the transport layer. Papers limited to high level traffic monitoring, protocol tuning or path measurement without direct packet inspection, or approaches that did not analyze TCP headers together with payloads were filtered out.

Theme labels were derived from the predominant TCP header fields and the analytic strategies reported across the corpus. These include port-based service mapping and policy enforcement (well-known, ephemeral, and reserved ranges), directionality and flow parsing via source and destination ports, connection-state tracking and misuse detection through flag logic (SYN, ACK, FIN, RST, and uncommon combinations), reliability and reassembly checks using sequence and acknowledgment numbers, header layout validation with the data offset and option bounds, flow-control assessment via the window field and window scale, capability and path-fit checks with options such as MSS, SACK, timestamps, and ECN, checksum verification with the pseudo-header for integrity across IPv4 and IPv6, and edge-case cues from the urgent pointer and reserved bits.

Figure 5 gives a visual snapshot of the TCP corpus as a word cloud built from the most frequent terms across the 32 included articles. The largest words, tcp, flow, service, port, packet, application, and classification, point to strong attention on port based service mapping, policy enforcement, and flow keyed inspection in traditional DPI. Connection and state terms, syn, ack, fin, rst, flag, handshake, and tracking, show a clear focus on following the TCP state machine and detecting misuse through unusual flag mixes. Frequent mentions of sequence, acknowledgment, window, and options indicate reliability checks, stream reassembly, and validation of flow control and capability hints. Words such as payload, parsing, rule, signature, and detection highlight header payload co inspection with deterministic matching. References to protocols and services, http, dns, ssh, telnet, and modbus, underline common TCP targets in anomaly studies. Taken together, the cloud anticipates the themes confirmed during coding, port based demultiplexing and policy checks, connection state and flag logic, ordering and flow control consistency, and application aware inspection within traditional DPI for TCP traffic.

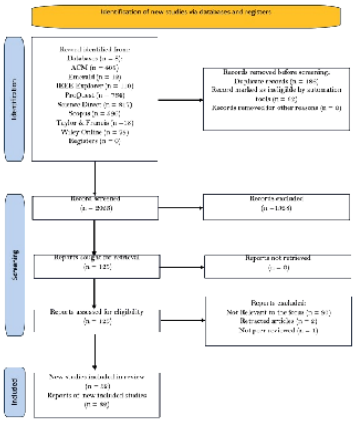

Figure 2: PRISMA Flow Diagram for Article Selection Process

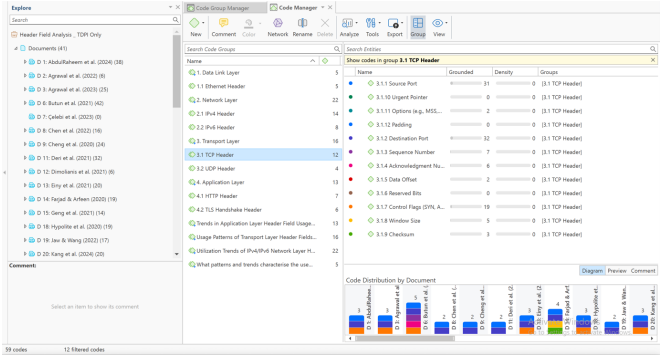

Figure 3: Established Code Group in ATLAS.ti

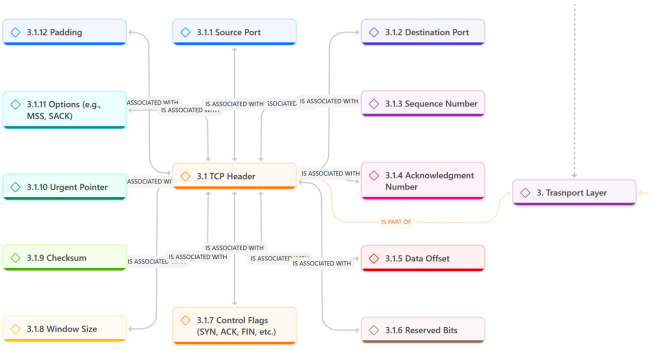

Figure 4: Thematic Code layer Taxonomy

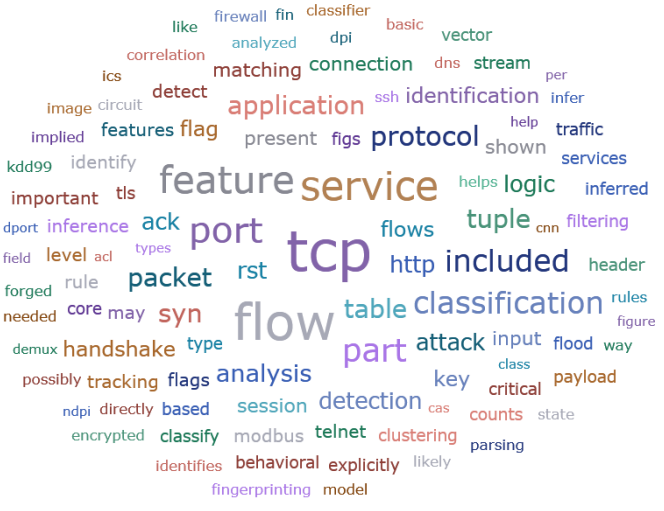

Figure 5: Word Cloud Generated from Articles

6. Results and Discussion

The main results of this thematic review on TCP header field usage in traditional DPI are presented in two parts. The quantitative analysis counts how many studies align with each theme, traces annual publication trends from 2020 to 2025, and charts publication activity by country and continent. The qualitative analysis then examines, theme by theme, how the selected studies apply Source Port, Destination Port, Sequence Number, Acknowledgment Number, Data Offset, Flags, Window Size, Checksum, Urgent Pointer, and common Options such as MSS, Window Scale, SACK, and Timestamps within rule based DPI, covering port based service mapping and policy enforcement, directionality and session parsing, connection state tracking and misuse detection through flag logic, reliability and ordering checks during reassembly, flow control assessment and capability signaling via options, and checksum validation across IPv4 and IPv6, all to support anomaly detection.

Viewed together, these quantitative and qualitative perspectives show the scale and global distribution of current work and clarify the core analytic approaches and emerging trends that characterize TCP header field usage in traditional DPI. They also indicate where attention is growing, for example stronger option parsing and normalization, tighter flag state consistency checks, more explicit handling of sequence gaps and overlaps, and increased use of checksum verification despite offload noise. At the same time, some usage remains limited or inconsistent, including sparse reporting on urgent pointer semantics, uneven treatment of window scaling edge cases, incomplete documentation of behaviors specific to IPv6, and limited discussion of NAT, asymmetric paths, and middlebox option stripping that can skew field level evidence.

7. Quantitative Findings

Within the five year review window from 2020 to 2025, 32 papers were identified that explicitly examine the use of TCP transport layer header fields in traditional, rule based DPI for anomaly detection, as summarized in Table 2. These studies span reputable journals and conference proceedings, yet they are unevenly distributed across outlets and years, which points to a specialized but steadily growing research area.

Of the 32 selected items, 16 appeared in conference proceedings and 16 in peer reviewed journals, a split that signals active scholarly work while confirming that the corpus focused specifically on TCP header field analysis within traditional DPI remains compact.

Table 1: Articles reviewed based on types of Journals

As shown in Figure 6, publication activity addressing TCP header fields in traditional DPI for anomaly detection increased early and then fluctuated. The corpus records 5 studies in 2020, 9 in 2021, 6 in 2022, 4 in 2023, 6 in 2024, and 2 in 2025. This sequence indicates a peak in 2021, a step down in 2022, a dip in 2023, a rebound in 2024, and a marked decline in 2025. Taken together, the counts suggest sustained but variable attention to TCP-header-aware, rule-based DPI for anomaly detection.

As shown in Figure 7, the global picture for research on TCP header-field usage in traditional DPI for anomaly detection is broad yet uneven. Country totals were compiled by examining all author affiliations in each article and counting each country once per publication, so multinational collaborations are credited to every participating country and yearly sums exceed the number of unique papers. In 2020, contributions come from five countries (notably China and Ukraine), totaling 7 country counts. Activity expands sharply in 2021 to 17 country counts, with participation spanning a wider mix of countries across Asia, Europe, and Oceania. Output contracts in 2022 to 9, concentrated within a smaller set of contributors, before continuing at a lower level in 2023 (5). The landscape broadens again in 2024 (10), reflecting renewed geographic spread, and then drops in 2025 (3), with only a few contributing countries recorded that year.

Across the full window, China contributes the most with 9 country counts, followed by the United States (6) and the Republic of Korea (4). India and Turkey contribute 3 each, while Malaysia, Pakistan, Poland, and Ukraine contribute 2 each; all remaining countries contribute 1. Overall, the distribution indicates multi-region participation, with visible year-to-year variability rather than sustained dominance by a single country..

Across the corpus(in network view Figure 11), the Source Port functions as a primary handle for traditional DPI, it anchors five-tuple flow keys, supports service inference and demultiplexing, and guides rule selection and session directionality. Studies use it for TCP flow and session identification or construction (Cheng et al., 2020; Ma et al., 2021; Butun et al., 2021; **?**; Moon et al., 2024; AbdulRaheem et al., 2024; Chen et al., 2022; Kang et al., 2024; **?**), for application and traffic classification or fingerprinting (Deri and Fusco, 2021; Liu et al., 2022; **?**; Agrawal et al., 2023; **?**; **?**; Mubarak et al., 2021a), and for correlating or screening misuse such as scans, floods, or login-attempt campaigns (Einy et al., 2021; Gulomov et al., 2025; Nam et al., 2023; Hypolite et al., 2020). Rule- and policy-oriented accounts echo this centrality, noting routine inclusion of source ports in signature rules and port-based enforcement (**?**Jaw and Wang, 2022; Nam et al., 2023). At the same time, port meaning is not always stable; several papers therefore pair source-port checks with destination-port context, handshake logic, and higher-level parsing to recover TCP context when mappings shift (Lee et al., 2022; Mubarak et al., 2021b; Nam et al., 2023)).

Theme 2: Destination Port

Across the corpus, the Destination Port(see Figure 12) is a central anchor for traditional DPI, it completes the five-tuple, supports service/protocol mapping and demultiplexing, and contributes to rule matching and policy-driven decisions (Ma et al., 2021; **?**; **?**; Liu et al., 2022; Nam et al., 2023). Studies use it for protocol and application classification, including explicit mappings to common services

Table 1
Articles reviewed based on types of journals/conferences.

Journal/Conference Name	2020	2021	2022	2023	2024	2025
ACM Conference	1	3	1	1	1	
Applied Sciences (Switzerland)				1		
Computer Networks					1	
Computer Systems Science and Engineering		1				
Connection Science			1			
Electronics (Switzerland)					1	
IEEE (CSR Conference)		1				
IEEE Access					1	
IEEE Conference	1	1	1	1		1
IEEE Conference (CCWC)			1			
IEEE Conference (TCSET)	1					
IEEE/ACM Transactions on Networking		1		1		
IEEE Trans. on Dependable and Secure Computing						1
International Journal of Information Technology (Singapore)					1	
International Journal of Information Technology and Web Eng.			1			
Mathematical Problems in Engineering		1				
PeerJ Computer Science			1			
Security and Communication Networks	1					
Sensors		1				
Sensors (Switzerland)	1					

such as HTTP (80) and TLS/HTTPS (443), and for protocol demux and protocol-type identification (Cheng et al., 2020; Liu et al., 2022; **?**; **?**; **?**). It also underpins rule-driven mechanisms such as port matching and redirection, and port-based policy rules (**?**Lee et al., 2022; Nam et al., 2023). Several papers further target ICS/OT identification using well-known destination ports (e.g., Modbus/TCP 502) or port sets to isolate industry protocols (Mubarak et al., 2021a,b). Destination-port signals are also used to interpret encrypted or non-standard-port contexts, including TLS over atypical ports and differentiation of encrypted applications (Deri and Fusco, 2021; Kang et al., 2024; **?**). Beyond direct matching, it appears as a feature for traffic classification, model training, and temporal modeling, and to refine tuple-based connection grouping (Agrawal et al., 2023; **?**; Chen et al., 2022; **?**; **?**). Tool-oriented accounts also report routine operational use of destination-port signals for service filtering and application/service inference (Jaw and Wang, 2022; AbdulRaheem et al., 2024).

In my interpretation, destination-port semantics are not always stable in operational networks due to multiplexing on popular ports and middlebox-driven remapping. Accordingly, robust analysis commonly pairs destination-port checks with complementary cues (e.g., source-port context, rule logic, and payload/semantic context) when classifying services or enforcing policy (Lee et al., 2022; Deri and Fusco, 2021; Nam et al., 2023).

Theme 3: Sequence Number

As proven in the Figure 13, the Sequence Number is a core cue for traditional DPI stream handling, supporting TCP reassembly and continuity reasoning. It is used for sequencing-anomaly detection and for tracking injection/teardown contexts, including RST-race and RST-focused packet-injection tracking (**?**Mubarak et al., 2021a; **?**; Nam et al., 2023). It also appears implicitly in handshake/state logic and in domain traces such as Modbus/TCP and Telnet (Butun et al., 2021; Mubarak et al., 2021b). Some papers reference it only as part of general TCP flow handling rather than as an explicit detection feature (Lee et al., 2022), underscoring its role as a supporting but important cue alongside other transport-layer indicators.

Theme 4: Acknowledgment Number

The Acknowledgment Number helps traditional DPI validate TCP session progression and supports connection and handshake verification as depicted in Figure 14. Studies use it in connection validation logic and to verify proper handshake flow (ACK-related checks) (**?**Mubarak et al., 2021b). It also appears in feature analysis where ACK is treated as an analytical field (Mubarak et al., 2021a). The field is further referenced in attack-related session manipulation contexts (Nam et al., 2023). In several accounts it is implied by the TCP handshake/state logic rather than modified directly, reflecting its supporting role alongside other transport-layer cues (Butun et al., 2021; Lee et al., 2022).

Theme 5: Data Offset

The Data Offset marks the TCP header length and the start of the payload, so traditional DPI uses it to anchor payload parsing and keep header interpretation aligned. As shown in Figure 15, Studies cite it explicitly for locating the payload and for correct TCP header parsing (**??**). In my interpretation, Data Offset can also serve as a practical sanity check in DPI parsers: values below the minimum header size, inconsistencies with option-length expectations, or offsets that exceed the segment size may indicate malformed traffic or evasion attempts. Even when handled implicitly

by parser libraries, validating the Data Offset helps avoid payload misalignment during content inspection.

Theme 7: Control Flags (SYN, ACK, FIN, etc.)

Control flags(see Figure 16) form the event grammar that traditional DPI uses to follow TCP state, validate handshakes, and detect misuse. Studies apply flag logic to start/teardown checks (3-way handshake, FIN/RST termination) and session validation (Butun et al., 2021; Lee et al., 2022; Hypolite et al., 2020), to behavioral/volume anomalies such as SYN-flood–like bursts (**?**) and broader connection-stage/attack-behavior interpretation (**?**), and to attack tracing including RST-injection and related manipulation contexts (Nam et al., 2023; Mubarak et al., 2021b). Flags are also used for policy enforcement and service logic in stateful controls (Liang and Kim, 2022). Many accounts treat flags as explicit features—binary indicators, counts, or feature-vector inputs, sometimes captured as time-based metadata or packet-sequence features (AbdulRaheem et al., 2024; Kang et al., 2024; Liu et al., 2022; Moon et al., 2024; **?**; **?**; Agrawal et al., 2023; Kim et al., 2021). A few papers note flags indirectly (e.g., inferred TCP status or hybrid-ML feature extraction) while still relying on them to orient session interpretation (Einy et al., 2021; Gulomov et al., 2025).

Theme 8: Window Size

The Window Size field is used in traditional DPI as a lightweight indicator that can be captured for analysis and profiling. Studies (see network view Figure 17) include the window value in processed datasets and feature extraction pipelines (Mubarak et al., 2021a; Liu et al., 2022), apply it as a clustering feature (Farjad and Arfeen, 2020), and reference concrete window observations in TCP flow traces (e.g., Win=2920) (Mubarak et al., 2021b). Several works also note potential diagnostic value in attack contexts, such as using atypical window behavior to help flag forged RST-related flow dynamics (**?**).

Window-size checks can be paired with sequence/ACK progress and handshake context to identify inconsistent advertised capacity or stalled peers; however, interpretation may depend on Window Scale and measurement conditions, so parsers often record the raw value and defer strict judgment to context-aware rules.

Theme 9: Checksum

The network view in Figure 18 of Checksum can be treated in traditional DPI as an integrity-related signal and is used only in a small subset of studies. In this corpus, it appears as part of clustering analysis (Farjad and Arfeen, 2020), is referenced in forged-packet inspection contexts (Nam et al., 2023), and is ensured during port rearrangement and reflection scenarios (Lee et al., 2022). From a practical perspective, checksum validation can function as a sanity check during parsing, but interpretation may be affected by capture and offload conditions; accordingly, many pipelines record checksum-related values and combine integrity checks with other transport cues rather than using checksum evidence alone.

Theme 11: Options (e.g., MSS, SACK, Timestamps)

TCP options appear infrequently in this corpus as presented in Figure 19, but they are surfaced in the analysis through MSS in SYN packets and as a recorded packet feature (Mubarak et al., 2021b; Liu et al., 2022). From a practical perspective, options can provide contextual capability signals (e.g., MSS/SACK) and may help flag malformed or implausible option settings; however, deployments typically treat option evidence cautiously and corroborate it with other transport cues rather than relying on options alone

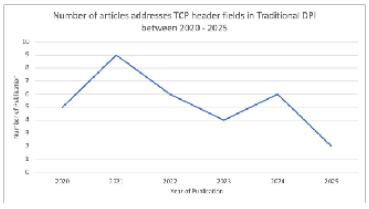

Figure 6: Breakdown of published articles according to the Year of Publication

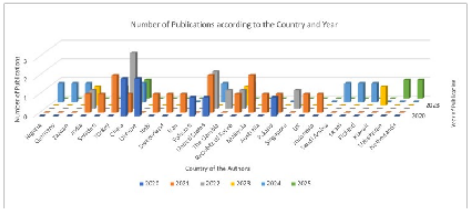

Figure 7: Breakdown of number of Publications According to Country and Year

Figure 8: Breakdown of Geographical map of authors

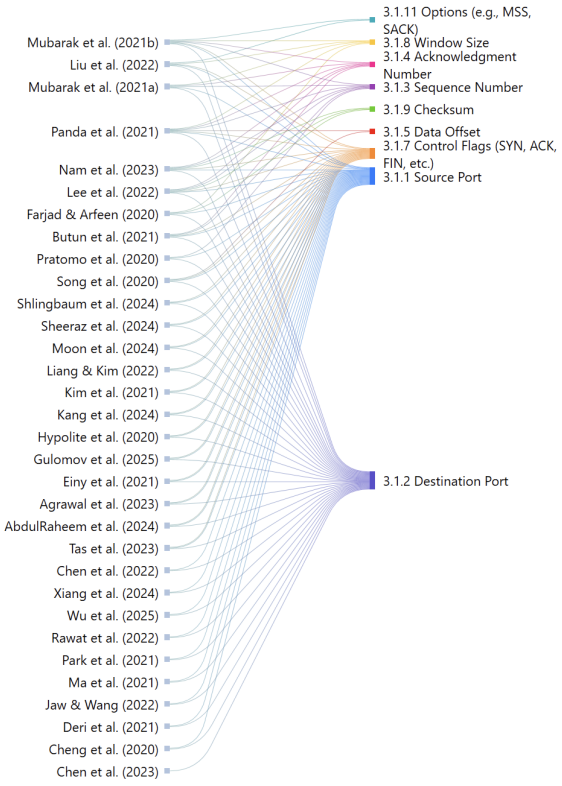

Figure 9: Thematic Mapping of the Publications

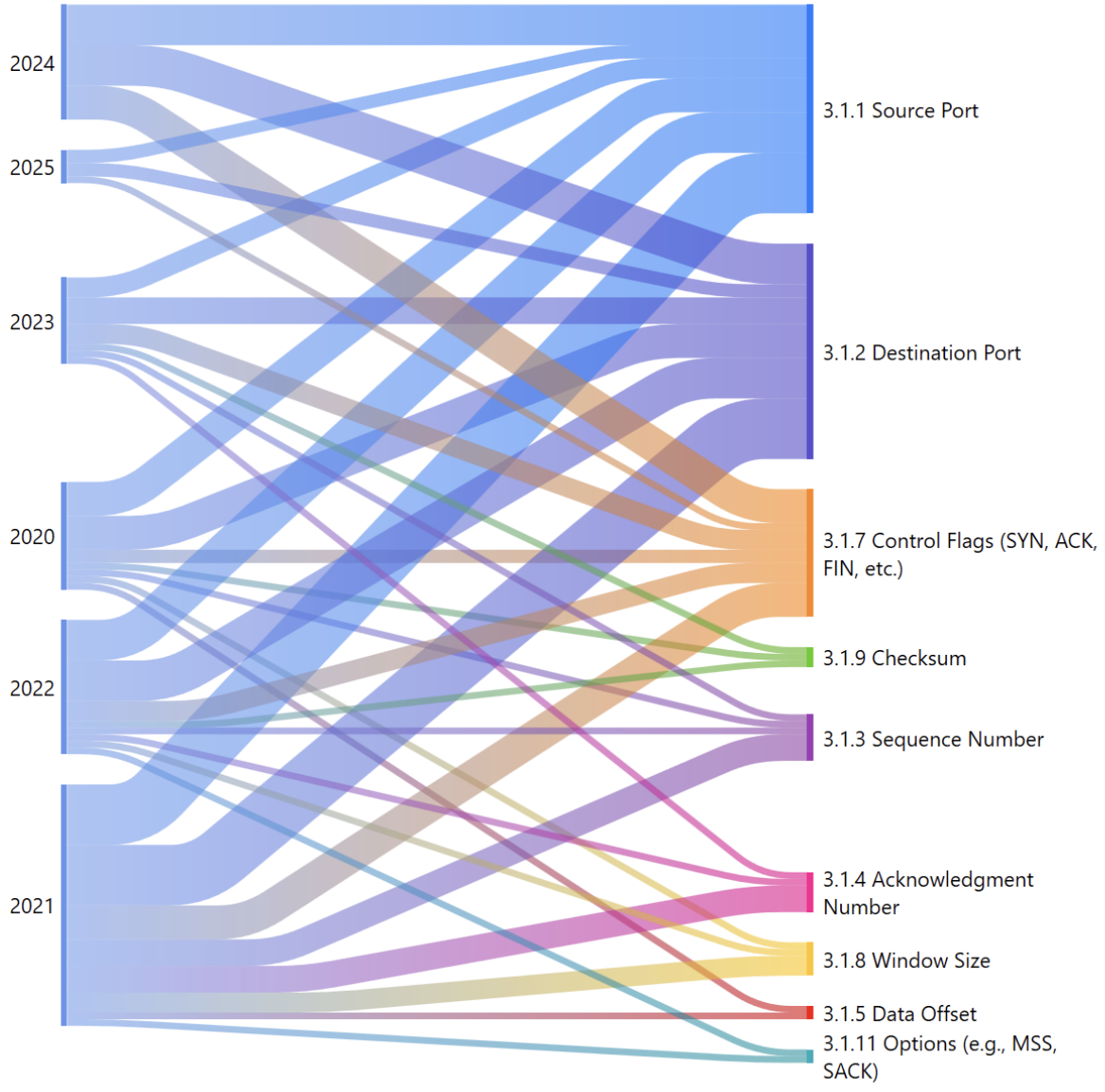

Figure 10: Sankey diagram of the Themes According to Year

Figure 11: Network View of the research articles on the "Source Port" Theme

Figure 12: Network View of the research articles on the "Destination Port" Theme

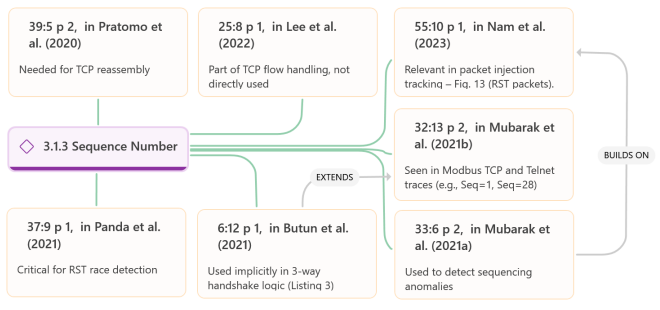

Figure 13: Network View of the research articles on the "Sequence Number" Theme

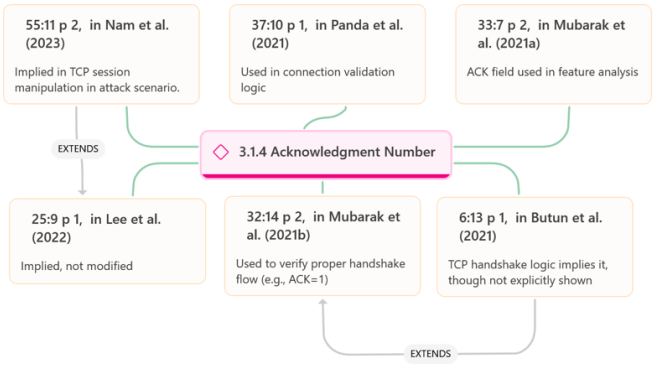

Figure 14: Network View of the research articles on the "Acknowledgment Number" Theme

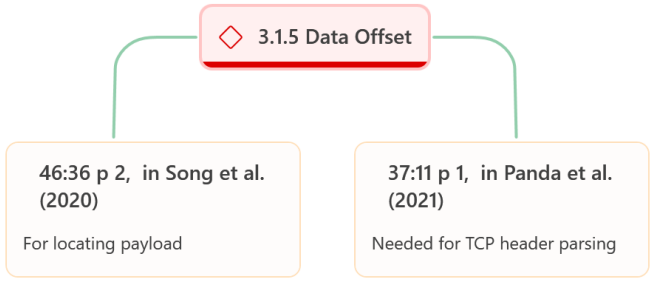

Figure 15: Network View of the research articles on the "Data Offset" Theme

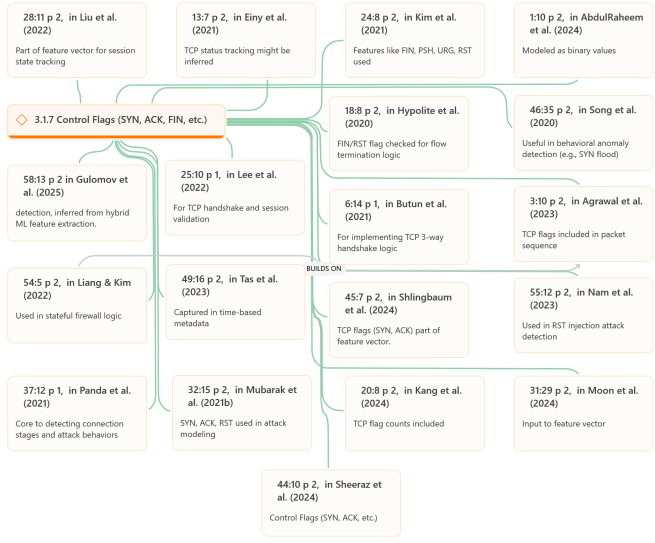

Figure 16: Network View of the research articles on the "Control Flags" Theme

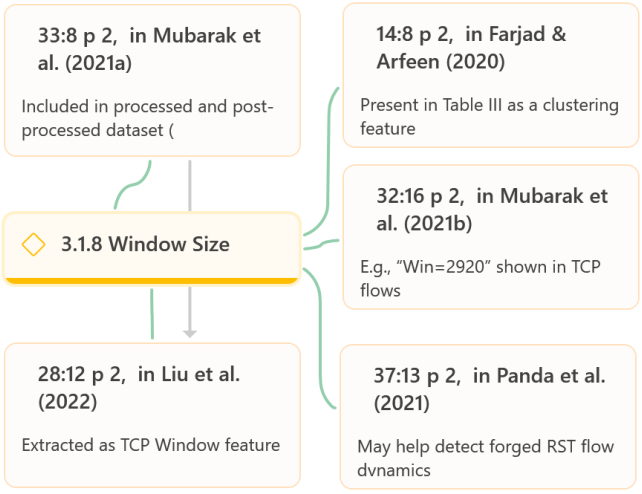

Figure 17: Network View of the research articles on the "Window Size" Theme

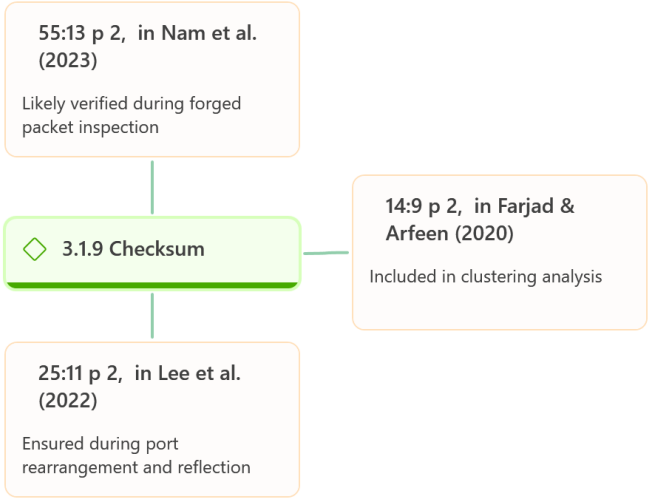

Figure 18: Network View of the research articles on the "Checksum" Theme

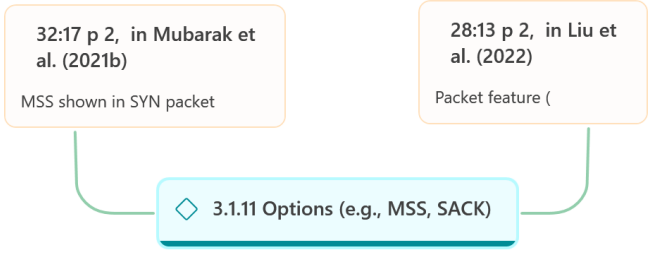

Figure 19: Network View of the research articles on the "Options" Theme

8. Results and Discussion

This review examines how TCP header fields are used in traditional DPI for anomaly detection from 2020 to 2025. Code-to-document mapping in ATLAS.ti and the aggregates in our tables show a tight, stable core of fields doing most of the practical work. Usage concentrates on Destination Port and Source Port for service mapping and rule selection, with Control Flags providing the state cues for handshake validation, teardown checks, and misuse screening. By comparison, Sequence and Acknowledgment numbers appear less often and mainly support reassembly, validation, and injection or manipulation tracking. Data Offset is used to align payload parsing. Window Size and Options are used infrequently as recorded features. Checksum appears in a small subset of studies rather than as a primary standalone detector. Notably, no included study in our coded 2020 to 2025 corpus explicitly analyzed the TCP Reserved bits, the Urgent Pointer, or header padding.

Reserved bits are meant to be zero and ignored, so any set values usually reflect buggy stacks or offload noise rather than signal. The Urgent Pointer is rarely used today and its semantics are uncertain, leaving datasets with little usable URG traffic and many tools suppress it. Header padding only aligns options, has no application meaning, and is typically normalized once the data offset is checked. Toolchains expose ports, flags, and sequence or acknowledgment numbers as primary features, so researchers favor those stable, interpretable cues. Put together, it is unsurprising that no study in 2020 to 2025 in our corpus used Reserved bits, the Urgent Pointer, or padding explicitly. Year by year the corpus is uneven but sustained: 5, 9, 6, 4, 6, and 2 papers across 2020 to 2025. The sequence suggests a maturing niche with an early rise and 2021 peak, a step down in 2022, a dip in 2023, a rebound in 2024, and a marked decline in 2025.

Ports act as the transport traffic switch, routing flows to the right parser or policy block, supporting endpoint directionality and flow keys, and anchoring rules for service recognition and misuse screening. From a practical perspective, limits recur under NAT, ephemeral reuse, proxies, and service multiplexing on popular ports, so robust pipelines pair port checks with payload and context. Control Flags carry the event grammar for traditional DPI. SYN, ACK, FIN, and RST drive connection tracking and expose anomalies such as incomplete handshakes, abnormal mixes, or reset injection. Their use peaks in 2021 and again in 2024 rather than increasing steadily, reflecting their role in fast, explainable triage. Sequence and Acknowledgment numbers reinforce ordering and integrity at the stream level. They help detect midstream interference and manipulation but are reported far less frequently than ports or flags. Data Offset guards header length and payload start. Window Size is recorded as a feature in a small subset of studies. Options surface occasionally, most often MSS in SYN packets, and Checksum is used sparingly. From a practical perspective, window, option, and checksum interpretation can be capture and implementation dependent, so pipelines often record these values and corroborate them with other transport cues rather than relying on them alone.

In summary, current traditional TCP DPI leans on ports for deterministic control and flags for state-aware screening, with sequence and acknowledgment supporting continuity and parser-alignment fields such as offset, window, options, and checksum appearing more selectively. Reserved bits, the Urgent Pointer, and header padding were not used explicitly in the reviewed studies. A constructive agenda is to make header extraction and validation steps explicit, publish baselines for flag sequences, handshake completion, window distributions, and option use, quantify robustness under NAT, proxies, asymmetric paths, middlebox option rewriting, and checksum offload, and document how port and flag gates steer payload inspection for faster, more selective anomaly detection.

9. Implications and Paths for Future Research

This review shows that traditional DPI for anomaly detection over TCP leans on a compact set of header cues. Destination and source ports steer service mapping and policy control, while control flags provide state cues for handshake validation, teardown checks, and rapid misuse screening. Sequence and acknowledgment numbers support ordering verification and reassembly, the data offset anchors payload parsing, window size appears as an infrequent recorded feature, options appear mainly as MSS evidence in SYN packets, and the checksum is used sparingly rather than as a primary detector. Reporting gaps are common in the included studies, including unclear header extraction steps, limited disclosure of reass0.5embly policies and overlap handling, sparse baselines for flag sequences and handshake completion, and thin discussion of option handling under normalization or middlebox rewriting. Practical capture factors such as NIC offload and mirroring artifacts are seldom discussed in relation to checksum interpretation, IPv6-specific behaviors are unevenly documented, and the effects of NAT, proxies, and asymmetric paths on field semantics are often left implicit. Reserved bits, the urgent pointer, and header padding were not explicitly analyzed in the reviewed studies and should be reported with clear presence rates and handling policies so that omissions are transparent.

10. Contributions and Benefits of the Study

This review maps how TCP header fields are used in traditional DPI for anomaly detection during 2020 to 2025 and organizes them into a compact field taxonomy. Core use centers on destination and source ports for service mapping and policy control, control flags for state tracking and misuse screening, sequence and acknowledgment numbers for continuity and reassembly support, and data offset for payload alignment. Window size, options, and checksum appear more selectively as recorded features or contextual checks, and reserved bits, the urgent pointer, and padding do not appear as explicitly analyzed fields in the corpus. The synthesis highlights gaps, including unclear extraction steps,

limited reporting on reassembly and overlap policies, sparse checksum and offload details, uneven IPv6 handling, thin option prevalence data, and missing baselines for flag sequences and handshake completion under NAT, proxies, and asymmetric routes. For researchers, these findings indicate where to invest next, including flag-sequence baselines, option handling transparency, checksum interpretation under capture artifacts, and robustness studies; for practitioners, the mapping links fields to concrete DPI tasks that support faster, more selective, and reproducible detection.

References

AbdulRaheem, M., Oladipo, I.D., Imoize, A.L., Awotunde, J.B., Lee, C.C., Balogun, G.B., Adeoti, J.O., 2024. Machine learning assisted snort and zeek in detecting ddos attacks in software-defined networking. International Journal of Information Technology 16, 1627–1643. doi:10.1007/s41870-023-01469-3.

Agrawal, A., Chatterjee, U., Maiti, R.R., 2023. Checkshake: Passively detecting anomaly in wi-fi security handshake using gradient boosting based ensemble learning. IEEE Transactions on Dependable and Secure Computing doi:10.1109/TDSC.2023.3236355. advance online publication.

Allman, M., Paxson, V., Stevens, W.R., 1999. TCP Congestion Control. Technical Report RFC 2581. RFC Editor. doi:10.17487/RFC2581.

Butun, I., Tuncel, Y.K., Oztoprak, K., 2021. Application layer packet processing using pisa switches. Sensors 21, 8010. doi:10.3390/s21238010.

Chen, J., Zhang, X., Wang, T., Zhang, Y., Chen, T., Chen, J., Xie, M., Liu, Q., 2022. Fidas: Fortifying the cloud via comprehensive fpga-based offloading for intrusion detection, in: Proceedings of the 49th Annual International Symposium on Computer Architecture (ISCA 2022), ACM. pp. 1–12. doi:10.1145/3470496.3533043.

Cheng, Z., Beshley, H., Urikova, O., Beshley, M., Kochan, O., 2020. Development of deep packet inspection system for network traffic analysis and intrusion detection. IEEE. Bibliographic details as supplied in the source document.

Dalkin, S.M., Greenhalgh, J., Jones, D., Cunningham, B., Lhussier, M., 2015. What's in a mechanism? development of a key concept in realist evaluation. Implementation Science 10, 49. doi:10.1186/s13012-015-0237-x.

Denyer, D., Tranfield, D., Van Aken, J.E., 2008. Developing design propositions through research synthesis. Organization Studies 29, 393–413. doi:10.1177/0170840607088020.

Deri, L., Fusco, F., 2021. Using deep packet inspection in cyber traffic analysis, in: 2021 IEEE International Conference on Cyber Security and Resilience (CSR), IEEE. pp. 1–6. doi:10.1109/CSR51186.2021.9527976.

Einy, S., Oz, C., Navaei, Y.D., 2021. The anomaly- and signature-based ids for network security using hybrid inference systems. Mathematical Problems in Engineering , 6639714doi:10.1155/2021/6639714. article 6639714.

Farjad, S.M., Arfeen, A., 2020. Cluster analysis and statistical modeling: A unified approach for packet inspection. Proceedings of the IEEE; ISBN 978-1-7281-6840-1.

Góngora-Blandón, M., Vargas-Lombardo, M., 2012. State of the art for string analysis and pattern search using cpu and gpu based programming. Journal of Information Security 3, 314–318. doi:10.4236/jis.2012.34038.

Gulomov, S.R., Khudayberganov, T.R., Turdiyev, T.T., Xasanov, A.B., Raximov, R.R., 2025. Hybrid traffic filtering and anomaly detection model for next-generation networks, in: 2025 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), IEEE. doi:10.1109/USBEREIT65494.2025.11054242.

Handley, M., Paxson, V., Kreibich, C., 2001. Network intrusion detection: Evasion, traffic normalization, and end-to-end protocol semantics, in: Proceedings of the 10th USENIX Security Symposium (USENIX Security 01), USENIX Association.

Hypolite, J., Sonchack, J., Hershkop, S., Dautenhahn, N., DeHon, A., Smith, J.M., 2020. Deepmatch: Practical deep packet inspection in the data plane using network processors, in: Proceedings of the 16th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '20), ACM. pp. 1–15. doi:10.1145/3386367.3431290.

Jaw, E., Wang, X., 2022. A novel hybrid-based approach of snort automatic rule generator and security event correlation (sarg-sec). PeerJ Computer Science 8, e900. doi:10.7717/peerj-cs.900.

Kang, H.S., Kim, K., Kim, S.R., 2024. A new mitigation method against dr-dos attacks using a snort udp module in low-specification fog computing environments. Electronics 13, 2919. doi:10.3390/electronics13152919.

Kim, J., Camtepe, S., Baek, J., Susilo, W., Pieprzyk, J., Nepal, S., 2021. P2dpi: Practical and privacy-preserving deep packet inspection, in: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), ACM. pp. 1–12. doi:10.1145/3433210.3437525.

Lee, J., Kang, M.S., Divakaran, D.M., Thet, P.M., Singhai, V., You, J.S., 2022. A step towards on-path security function outsourcing, in: Proceedings of the International Conference on Distributed Computing and Networking (ICDCN'22), ACM. p. 13. doi:10.1145/nnnnnnn.nnnnnnn. the source document supplied DOI 10.1145/nnnnnnn.nnnnnnn; this DOI should be verified before submission.

Liang, J., Kim, Y., 2022. Evolution of firewalls: Toward securer network using next generation firewall, in: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), IEEE. pp. 1–7. doi:10.1109/CCWC54503.2022.9720922.

Liu, X., Liu, Z., Zhang, Y., Zhang, W., Lv, D., Zhou, Q., 2022. Tcn enhanced novel malicious traffic detection for iot devices. Connection Science 34, 1322–1341. doi:10.1080/09540091.2022.2067124.

Ma, C., Chen, S., Zhang, Y., Xiao, Q., Odegbile, O.O., 2021. Super spreader identification using geometric-min filter. IEEE/ACM Transactions on Networking 30, 112–126. doi:10.1109/TNET.2021.3108033.

Moon, S.J., Srivastava, M., Bieri, Y., Martins, R., Sekar, V., 2024. Pryde: A modular generalizable workflow for uncovering evasion attacks against stateful firewall deployments, in: Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), IEEE. pp. 1–12. doi:10.1109/SP54263.2024.00144.

Mubarak, S., Habaebi, M.H., Islam, M.R., Abdul Rahman, F.D., Tahir, M., 2021a. Anomaly detection in ics datasets with machine learning algorithms. Computer Systems Science and Engineering 37, 33–46. doi:10.32604/csse.2021.014384.

Mubarak, S., Habaebi, M.H., Islam, M.R., Khan, S., 2021b. Ics cyber-attack detection with ensemble machine learning and dpi using cyber-kit datasets, in: Proceedings of IEEE Conference, IEEE. pp. 1–6.

Nam, J., Lee, S., Porras, P., Yegneswaran, V., Shin, S., 2023. Secure inter-container communications using xdp/ebpf. IEEE/ACM Transactions on Networking 31, 672–685. doi:10.1109/TNET.2022.3206781.

Noyes, J., Popay, J., Pearson, A., Hannes, K., Booth, A., 2008. Qualitative research and cochrane reviews, in: Higgins, J., Green, S. (Eds.), Cochrane Handbook for Systematic Reviews of Interventions. Wiley. volume 5.0.1, pp. 1–18.

Panda, S., Feng, Y., Kulkarni, S.G., Ramakrishnan, K.K., Duffield, N., Bhuyan, L.N., 2021. Smartwatch: Accurate traffic analysis and flow-state tracking for intrusion prevention using smartnics, in: Proceedings of the 17th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '21), ACM. pp. 1–12. doi:10.1145/3485983.3494861.

Park, T., Nam, J., Na, S.H., Chung, J., Shin, S., 2021. Reinhardt: Real-time reconfigurable hardware architecture for regular expression matching in dpi, in: ACSAC '21: Annual Computer Security Applications Conference, ACM. pp. 1–14. doi:10.1145/3485832.3485878.

Pawson, R., Greenhalgh, T., Harvey, G., Walshe, K., 2005. Realist review—a new method of systematic review designed for complex policy interventions. Journal of Health Services Research & Policy 10, 21–34. doi:10.1258/1355819054308530.

Paxson, V., 1998. Bro: A system for detecting network intruders in real time, in: Proceedings of the 7th USENIX Security Symposium, San Antonio, Texas. URL: https://www.usenix.org/legacy/publications/

library/proceedings/sec98/full_papers/paxson/paxson.pdf.

Postel, J., 1981. Transmission Control Protocol. Technical Report RFC 793. RFC Editor. doi:10.17487/RFC0793.

Pratomo, B.A., Burnap, P., Theodorakopoulos, G., 2020. Blatta: Early exploit detection on network traffic with recurrent neural networks. Security and Communication Networks , 8826038doi:10.1155/2020/8826038. article 8826038.

Rawat, R., Garg, B., Pachlasiya, K., Mahor, V., Telang, S., Chouhan, M., Shukla, S.K., Mishra, R., 2022. Scnta: Monitoring of network availability and activity for identification of anomalies using machine learning approaches. International Journal of Information Technology and Web Engineering 17, 1–15. doi:10.4018/IJITWE.297971.

Sheeraz, M., Durad, M.H., Tahir, S., Tahir, H., Saeed, S., Almuhaideb, A.M., 2024. Advancing snort ips to achieve line rate traffic processing for effective network security monitoring. IEEE Access 12, 61848–61859. doi:10.1109/ACCESS.2024.3395123.

Shlingbaum, E., Ben Yehuda, R., Kiperberg, M., Zaidenberg, N.J., 2024. Virtualized network packet inspection. Computer Networks 251, 110619. doi:10.1016/j.comnet.2024.110619.

Song, W., Beshley, M., Przystupa, K., Beshley, H., Kochan, O., Pryslupskyi, A., Pieniak, D., Su, J., 2020. A software deep packet inspection system for network traffic analysis and anomaly detection. Sensors 20, 1637. doi:10.3390/s20061637.

Tas, I.M., Baktir, S., 2023. A novel approach for efficient mitigation against the sip-based drdos attack. Applied Sciences 13, 1864. doi:10.3390/app13031864.

Wu, P., Ning, J., Huang, X., Chen, R., Zhang, K., Liang, K., 2025. Privbox: Privacy-preserving deep packet inspection with dual double-masking obfuscated rule generation. IEEE Transactions on Dependable and Secure Computing 22. doi:10.1109/TDSC.2025.3557423.

Xiang, J., Zhang, X., Zheng, Q., Deng, L., Zhao, D., Zhou, J., 2024. Cidf: Combined intrusion detection framework in industrial control systems based on packet signature and enhanced fsfdp, in: Proceedings of the 15th Asia-Pacific Symposium on Internetware (Internetware 2024), ACM. pp. 1–10. doi:10.1145/3671016.3674812.