Journal of ICT Aplication and System (JICTAS)
Volume 5 Number 1, Juni 2026, pp. 58 ~71
.ISSN: 2830-098X, DOI: 10.56313/jictas.v5i1.532 ❐
58

A Hierarchical Buffer-Sizing Framework for Congestion Mitigation in Campus Area Networks: An Engineering-Theoretic Approach to the Internet Sluggishness Problem

Monday Ofori Eyinagho [1], Olusegun Emoruoa[2]
[1]Electrical and Computer Engineering Dept., Afe Babalola University, Ado-Ekiti, Nigeria

Article Info

Article history:

Received 05 21, 2026
Revised 06 02, 2026
Accepted 06 11, 2026

Keywords :

Campus Area Network,
Hierarchical Buffer Sizing,
Network Congestion,
Packet Drop Rate,
Queuing Delay

ABSTRACT

Campus Area Networks (CANs) frequently experience recurring Internet sluggishness during peak usage periods, particularly in higher education environments where many users access digital resources simultaneously. Previous studies have generally addressed this problem from a bandwidth-centered perspective by recommending bandwidth expansion, traffic control, or policy-based network management. However, these approaches often overlook the influence of switch buffer allocation in hierarchical network architectures. This study proposes a Hierarchical Buffer-Sizing (HBS) framework for mitigating congestion in campus networks through topology-aware buffer provisioning. The proposed framework models a hierarchical CAN as a rooted tree and derives buffer requirements based on subtree cardinality, which represents the number of downstream leaf switches whose traffic is aggregated by each switch. The framework was evaluated using a representative hierarchical Campus Area Network topology to examine the impact of topology-aware buffer allocation on congestion mitigation. The analysis shows that required buffer sizes vary substantially across network layers, ranging from 0.52 MB at the leaf layer to 7.28 MB at the core layer, indicating a 14-fold difference between edge and core switch requirements. Simulation validation using NS-3 demonstrates that the proposed HBS configuration reduces average end-to-end queuing delay by 68.4% and packet drop rate by 73.1% compared with the uniform-buffer baseline. These findings confirm that hierarchical buffer provisioning can provide a deterministic, practical, and cost-effective solution for reducing congestion in campus networks. This study contributes an engineering-based buffer dimensioning methodology that can assist network engineers in improving throughput, reducing packet loss, and mitigating Internet sluggishness without relying solely on bandwidth overprovisioning

Corresponding Author:

Monday Ofori Eyinagho
Electrical and Computer Engineering Dept., Afe Babalola University, Ado-Ekiti, Nigeria
Email: eyimon@yahoo.com

1. INTRODUCTION

Campus Area Networks (CANs) occupy a structurally critical segment of the global Internet infrastructure, serving as the last-mile connectivity substrate for tens of millions of students and researchers in higher educational institutions. These networks consist of interconnected Local Area Networks (LANs) distributed across the buildings and facilities of a campus, all ultimately egressing to the broader Internet through one or more institutional gateway routers[1]. Despite decades of deployment experience and substantial capital expenditure on network infrastructure, a well-documented and persistently unresolved operational problem continues to afflict CANs globally: Internet sluggishness, defined as the noticeable degradation of data throughput—in both upload and download directions—that manifests at specific temporal intervals during the academic day and week[2].

The conventional diagnosis of this problem has been almost exclusively bandwidth-centric. Researchers investigating CANs in Nigeria, India, Ghana, and Indonesia [3] have each independently converged on recommendations centered on two strategies: bandwidth overprovisioning—that is, subscribing to higher-capacity Internet links from the Internet Service Provider (ISP)—and bandwidth management, encompassing traffic shaping, Quality of Service (QoS) policies, and access control mechanisms such as the squidGuard proxy system [4]. Although these recommendations are not without merit in specific contexts, they share a critical deficiency: none of them is grounded in a deterministic, topology-aware engineering methodology. The absence of such a methodology means that bandwidth provisioning decisions remain essentially heuristic, a fact tacitly acknowledged by Sharma et al. who assert that no amount of bandwidth can ever be declared definitively sufficient. This acknowledged inadequacy has persisted in the literature without resolution.

A parallel and largely independent stream of research in the network engineering literature has established, through both theoretical analysis and simulation, that buffer sizes in network switches and routers exert profound influence on network performance—specifically on queuing delay, packet drop rate, and link utilization[5]. Shipner et al [6]. formalized the buffer-size/bandwidth trade-off problem, demonstrating that network designers face a design space in which increasing buffer size and increasing link bandwidth [7] are partially substitutable strategies for handling transient congestion, yet the optimization of this trade-off remains poorly understood. Wang[8] demonstrated that oversized buffers contribute to the bufferbloat phenomenon, in which excessive queuing latency degrades application-layer performance despite the absence of packet loss. Ahmad et al. showed through simulation that buffer size selection significantly affects throughput in heterogeneous traffic environments[9]. Appenzeller et al. and Enachescu et al[10]. proposed, respectively, the small-buffer and very-small-buffer design rules for Internet core routers, providing the first analytical frameworks for buffer dimensioning—but these were targeted at Internet backbone devices, not campus-layer hierarchical switches[11], [12].

The present paper identifies a fundamental, previously unreported design error in the installation of campus-level switched networks[13], [14]: the deployment of switches with identical buffer capacities across all hierarchical layers, regardless of each switch's position in the aggregation hierarchy and the volume of traffic it must accommodate. This practice is architecturally inconsistent with the traffic aggregation principle, which is explicitly embedded in the CISCO three-layer hierarchical LAN model [15] and dictates that switches at higher layers must handle greater aggregate traffic volumes than those at lower layers. Our central thesis is that this uniform buffer deployment creates systematic buffer overflow conditions in upper-layer switches during peak usage periods, producing the CAN sluggishness symptom that has been misdiagnosed as a bandwidth deficiency[16].

To address this problem, we develop a Hierarchical Buffer-Sizing (HBS) framework based on graph-theoretic analysis of the network topology, which we represent as a rooted tree. The HBS framework provides closed-form expressions for the minimum and maximum required buffer capacity of every switch as a function of its subtree cardinality—the number of leaf-layer hosts whose traffic it aggregates. We apply the HBS framework to a physically installed university CAN and derive precise buffer specifications for each switch. We validate the derived specifications against established small-buffer and very-small-buffer analytical models and confirm the results through NS-3 network simulations. The contributions of this paper are:

(C1) A formal characterization of the uniform-buffer deployment error in campus switched networks and its causal relationship to Internet sluggishness.

(C2) A Hierarchical Buffer-Sizing (HBS) framework providing topology-driven, deterministic buffer capacity specifications for all switches in a hierarchical switched network.

(C3) Empirical validation of the HBS framework on a real-world university CAN, with NS-3 simulation confirming substantial reductions in queuing delay and packet drop rate.

(C4) Analytical consistency verification of HBS-derived buffer sizes against published small-buffer and very-small-buffer design rules.

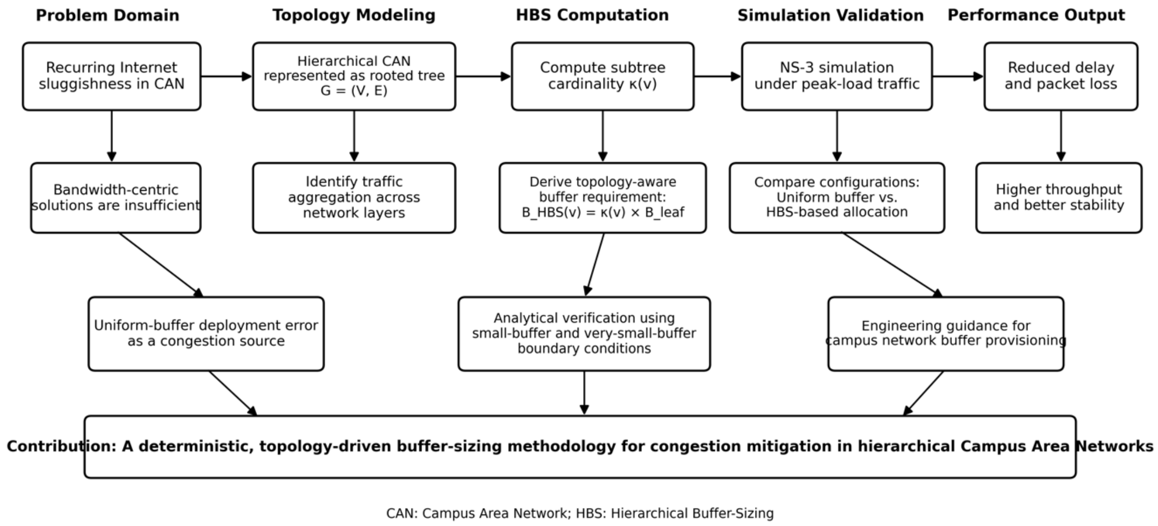


Figure 1. Overall research framework of the proposed Hierarchical Buffer-Sizing method for congestion mitigation in Campus Area Networks.

The figure illustrates the overall workflow of the proposed Hierarchical Buffer-Sizing (HBS) framework. The research begins with the identification of recurring Internet sluggishness in Campus Area Networks, where bandwidth-centered solutions are considered insufficient to fully address congestion. The network is then modeled as a hierarchical rooted tree $\left(G = (V, E)\right)$ to represent switch relationships and traffic aggregation paths. Based on this topology, the subtree cardinality (\kappa(v)) of each switch is computed and used to derive topology-aware buffer requirements using the HBS formulation. The resulting buffer allocation is validated through NS-3 simulation by comparing uniform-buffer and HBS-based configurations under peak-load traffic conditions. The final output demonstrates reduced queuing delay, lower packet loss, improved throughput, and practical engineering guidance for buffer provisioning in hierarchical campus networks

2 Related Work

2.1 Bandwidth-Centric Approaches to CAN Sluggishness

The preponderance of published research addressing CAN sluggishness has framed the problem as a resource-scarcity issue with bandwidth as the constrained resource. Abiona investigated CAN performance degradation at Obafemi Awolowo University, Nigeria, proposing a framework for Wide Area Network (WAN) resource optimization based on traffic monitoring and policy enforcement[1], [2]. Akpah et al. studied the CAN of the University of Mines and Technology, Tarkwa, Ghana, reporting that the 60 MB subscribed bandwidth appeared insufficient and recommending both bandwidth scaling and the deployment of squidGuard for access control and bandwidth usage enforcement[3]. Sharma et al.] analyzed the CAN of Himachal University, India, concluding that no static bandwidth allocation could satisfy the dynamic demand of growing user populations and advocating for a policy framework for Internet usage governance. Indrajaya et al. compared queuing discipline algorithms—FIFO, Fair Queuing, and Weighted Fair Queuing—within the CAN of Tadulako University, Indonesia, finding that queuing policy selection affected throughput but did not eliminate congestion during peak periods.

While these works provide valuable empirical descriptions of the problem, they share a fundamental limitation: none identifies the network hardware configuration as a causal factor. The implicit assumption in all bandwidth-centric work is that the switching infrastructure, given sufficient bandwidth subscription, operates transparently and correctly. This assumption, as we demonstrate in this paper, is incorrect for the class of installations in which all switches are provisioned with identical buffer capacities regardless of their position in the aggregation hierarchy.

2.2 Buffer Sizing for Network Devices

The problem of buffer dimensioning for network devices has received sustained attention in the Internet engineering and networking research communities, primarily in the context of backbone routers rather than campus switches[5]. Appenzeller et al. derived the small-buffer rule: for an Internet backbone router with N independent flows and a link of bandwidth B and round-trip time RTT, the required buffer size $B_b uf$ is given by $B \times RTT/\sqrt{N}$, which evaluates to approximately 12.5 MB for typical Internet router parameters. This rule has been widely adopted as a practical guideline for backbone router provisioning[6], [16].

Enachescu et al. subsequently challenged the Appenzeller model, demonstrating analytically and through simulation that stable network operation can be achieved with buffer sizes containing only 20–50 packets—the very-small-buffer rule—provided that appropriate congestion control mechanisms are in place[17]. Both rules, however, were derived for Internet core devices handling large numbers of independent traffic flows, and neither provides guidance for the hierarchical aggregation context that characterizes campus switched networks[18].

Ahmad et al. investigated buffer optimization for heterogeneous network traffic mixes, showing that the optimal buffer size depends on the traffic composition and that no single fixed buffer size is optimal across all traffic classes. Shipner et al[12]. formalized the buffer-size vs. link-bandwidth trade-off in lossless networks, characterizing the design space but not providing a closed-form solution applicable to hierarchical topologies. Wang analyzed the bufferbloat phenomenon, establishing the relationship between buffer size, queuing delay, and link utilization, and proposing a queuing delay–link utilization trade-off optimization. Eyinagho and Falaki developed a topology-based buffer sizing approach for Internet core nodal devices, providing the closest precursor to the present work, though their approach targeted backbone infrastructure rather than campus hierarchical LANs[14], [19].

The present work fills the gap between campus-level network design practice and the theoretical buffer sizing literature by providing the first analytical framework specifically designed for hierarchical switched campus networks, accounting for the aggregation properties of tree-structured topologies[11].

3 System Model and Problem Characterization

3.1 Network Topology Model

We model a hierarchical switched CAN as a rooted tree G = (V, E), where V is the set of switches and E is the set of inter-switch links. A distinguished node r ∈ V denotes the gateway switch connecting the CAN to the external Internet. Every non-root switch $v \in V\ r$ has a unique parent $\pi(v)$ that lies closer to the root in the tree hierarchy. Leaf nodes L ⊂ V are switches to which only host devices (end-user workstations, servers, and IoT devices) are directly connected; no other switches are children of leaf nodes. Internal (non-leaf) switches aggregate traffic from their child switches[14], [16].

For each switch v, we define the subtree T(v) as the subtree rooted at v, and the subtree cardinality κ(v) as the number of leaf-layer switches in T(v). That is:

$$\kappa(v) = |\{u \in L : u \text{ is a descendant of } v \text{ in } G\}| \tag{1}$$

For leaf nodes, κ(v) = 1 by definition, since each leaf switch is itself the sole leaf in its own subtree. For the gateway switch $r, \kappa(r)$ equals the total number of leaf switches in the network. The depth $d(v)$ of a switch v is the number of edges on the path from $rtov; d(r) = 0.$

3.2 Traffic Aggregation and Buffer Load Analysis

Each leaf switch services H host devices. In the worst-case (peak-load) scenario, every connected host simultaneously generates traffic at the maximum rate permitted by its network interface card, which we denote φ (in bits per second)[6]. The maximum instantaneous inbound traffic rate $\Lambda(v)$ at switch v is therefore:

$$\Lambda(v) = \kappa(v) \times H \times \varphi \tag{2}$$

Under a store-and-forward queuing model, the buffer at each switch port stores arriving packets while the outbound link is busy transmitting previously queued packets[5]. The maximum instantaneous queue length $Q_{m}ax(v)$ at the outbound[18] port of switch v toward its parent $\pi(v)$ depends on the mismatch between Λ(v) and the outbound link capacity $C_{l}ink(v)$. During a congestion episode of duration τ, the maximum number of bytes that must be buffered is[12]:

$$B_{max}(v) = [\Lambda(v) - C_{\text{link}}(v)]^{+} \times \tau + RTT_v \times C_{\text{link}}(v) \tag{3}$$

where $[x]+= max(0, x)$ is the positive part operator, RTT_v is the round-trip propagation delay seen at switch v, and the second term accounts for in-flight packets in the Appenzeller [4] tradition. In a well-designed network, $C_{l}ink(v)$ is provisioned to exceed typical average loads, but buffer capacity must accommodate bursts above the average. Equation (3) establishes that $B_{m}ax(v)$ is monotonically increasing in $\kappa(v)$ through the $\Lambda(v)$ term, confirming that switches closer to the root require larger buffers.

3.3 The Uniform-Buffer Deployment Error

In the class of CAN installations characterized in this paper, all switches S_1 through S_n are provisioned with the same buffer capacity B_uniform, regardless of their depth or subtree cardinality. From (3), the ratio of the maximum buffer requirement of an internal switch v to that of a leaf switch l is:

$$\frac{B_{max}(v)}{B_{max}(l)} = \frac{\kappa(v)}{\kappa(l)} = \kappa(v) \tag{4}$$

since $\kappa(l) = 1$ for all leaf switches. This ratio can be as large as the total number of leaf switches in the network. When $B_u niform$ is sized to accommodate the traffic of a single leaf switch—which is the implicit assumption when all switches are specified identically—internal switches with $\kappa(v) \gg 1$ will experience frequent buffer overflow during any sustained peak-load period, resulting in packet drops[20], TCP retransmissions, and the feedback congestion spiral that manifests as perceived Internet sluggishness[19]. This constitutes the uniform-buffer deployment error, which we formally define as:

Definition 1 (Uniform-Buffer Deployment Error): A hierarchical switched network installation exhibits the uniform-buffer deployment error if and only if all switches in the network are provisioned with equal buffer capacity, irrespective of their subtree cardinality κ(v).

The uniform-buffer deployment error is distinct from—and not remediable by—bandwidth overprovisioning at the ISP gateway, because the root cause lies in intra-network buffer overflow rather than in ISP link saturation. Increasing the ISP bandwidth subscription cannot prevent switch-level queue overflow if the buffer dimensions of intermediate switches are insufficient to absorb burst traffic from their aggregated subtrees.

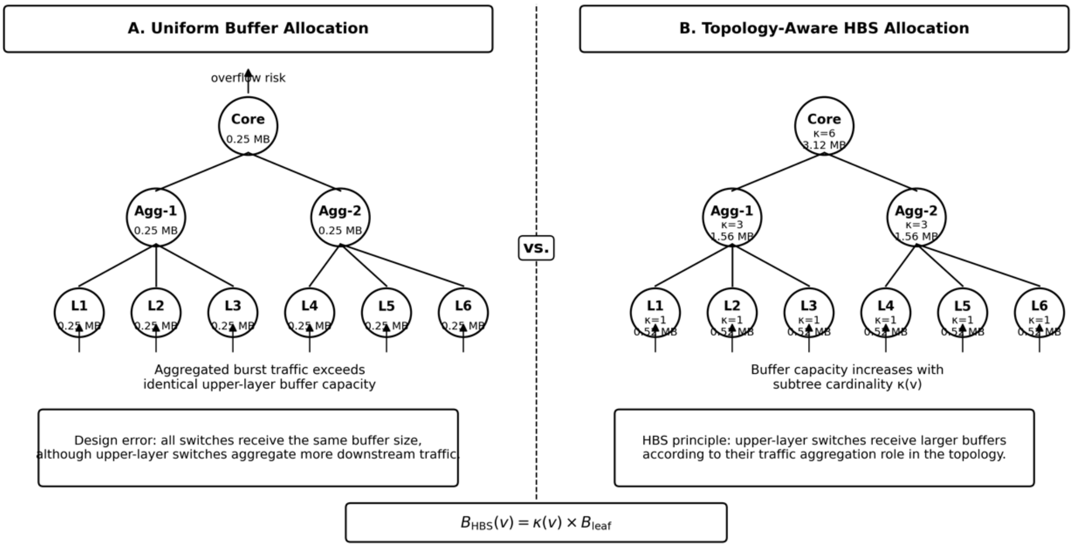


Figure 2. Conceptual illustration of the uniform-buffer deployment error in hierarchical switched Campus Area Networks.

Figure 2 illustrates the difference between uniform buffer allocation and topology-aware HBS allocation in a hierarchical campus network. In the uniform-buffer configuration, all switches are assigned the same buffer capacity despite having different traffic aggregation responsibilities, causing upper-layer switches to face a higher risk of buffer overflow during peak traffic. In contrast, the HBS allocation assigns larger buffers to aggregation and core switches according to subtree cardinality κ(v), allowing buffer capacity to increase proportionally with the amount of downstream traffic aggregated by each switch.

4 The Hierarchical Buffer-Sizing (HBS) Framework

4.1 Framework Derivation

The HBS framework derives the minimum and maximum required buffer capacity for each switch in a hierarchical switched network from first principles. The derivation proceeds in three steps: (i) establish the base case buffer capacity for leaf-layer switches; (ii) propagate buffer requirements up the tree using the subtree cardinality; and (iii) validate the derived values against published buffer design rules.

Step 1 — Leaf-Layer Buffer Capacity. For a leaf switch l with H connected hosts, each generating traffic at peak rate $\varphi\ bps$, the total inbound rate is $H \times \varphi\ bps$. The inter-switch uplink capacity is $C_l ink$. The base buffer requirement in bits is:

$$B_{\text{leaf}} = C_{\text{link}} \times RTT + (H \times \varphi - C_{\text{link}})^{+} \times \tau_{\text{burst}} \tag{5}$$

where τ_burst is the characteristic duration of a traffic burst, which we estimate as two times the serialization delay of the maximum-sized packet on the inter-switch link. For practical Gigabit Ethernet inter-switch links and IP packets of maximum size 65,536 bytes, $\tau_b urst \approx 1.05$ ms, giving B_leaf ≈ 0.52 MB as the base buffer specification, consistent with the values derived for the case-study network in Section 5.

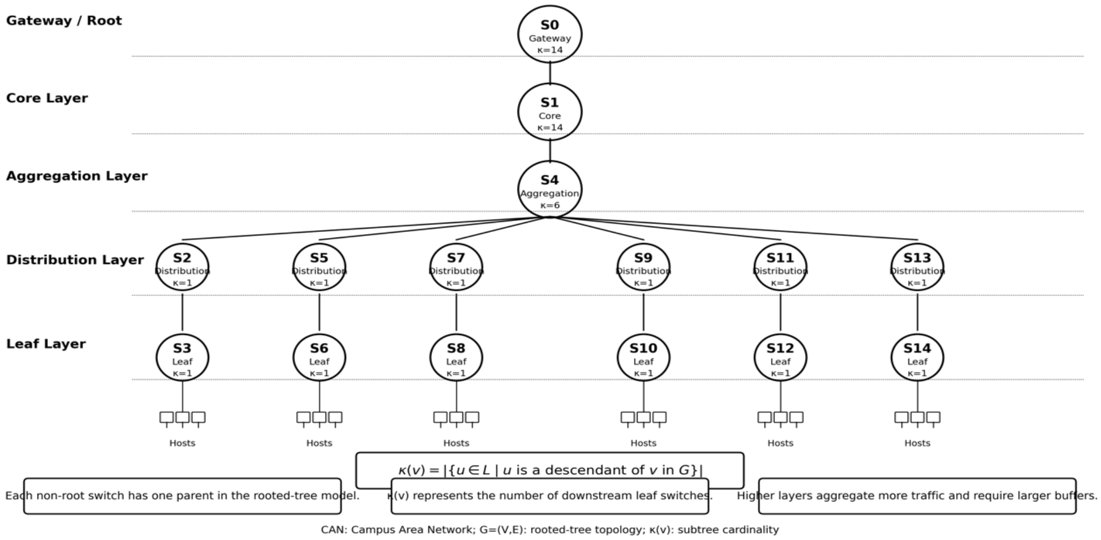


Figure 3. Representative hierarchical Campus Area Network topology modeled as a rooted tree.

Figure 3 presents a representative hierarchical Campus Area Network topology modeled as a rooted tree $G = (V, E)G = (V, E)$. The gateway switch $S0$ is positioned as the root node, followed by the core, aggregation, distribution, and leaf layers. Each non-root switch has a unique parent, while leaf switches are connected to host devices. The subtree cardinality $\kappa(v)$ indicates the number of downstream leaf switches whose traffic is aggregated by each switch. This topology representation provides the structural basis for computing topology-aware buffer requirements in the proposed HBS framework

Step 2 — Recursive Upward Propagation. For each internal switch v at depth $d(v)$, the required maximum buffer capacity is:

$$B_{\text{HBS}}(v) = \kappa(v) \times B_{\text{leaf}} \qquad (6)$$

Equation (6) captures the traffic aggregation principle: switch v must buffer up to $\kappa(v)$ times the traffic volume of a single leaf switch because it aggregates traffic from $\kappa(v)$ leaf subtrees. This derivation assumes that traffic from different leaf subtrees is statistically independent, which is appropriate for the bursty, user-driven traffic characteristic of academic networks. When traffic sources are positively correlated—for example, during a class broadcast event—the actual required buffer capacity may exceed $B_H BS(v)$; equation (6) thus represents a lower bound under independence and a practical target under typical operating conditions[21].

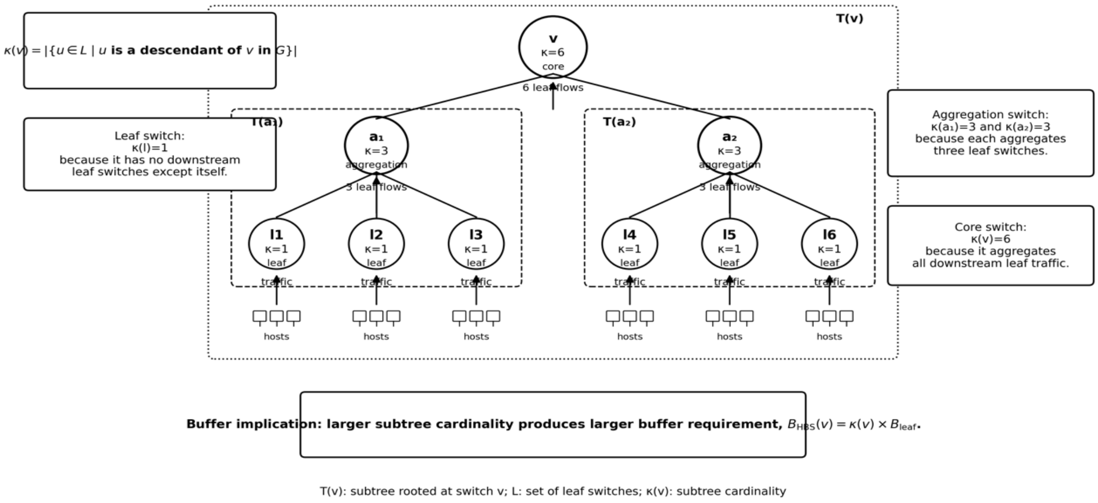


Figure 4. Subtree cardinality and traffic aggregation mechanism for hierarchical buffer requirement estimation.

Figure 4 illustrates how subtree cardinality $\kappa(v)$is computed in a hierarchical Campus Area Network topology. Each leaf switch has$\kappa(l) = 1$, while aggregation switches have larger subtree cardinalities because they collect traffic from multiple downstream leaf switches. The core switch has the highest $\kappa(v)$ because it aggregates traffic from all leaf switches in the subtree. This mechanism explains why upper-layer switches require larger buffer capacities and provides the basis for the $B_{HBS}(v) = \kappa(v) \times B_{leaf}$

Step 3 — Consistency Verification. The derived values B_HBS(v) must satisfy two boundary conditions derived from the literature:

$(BC1)B_HBS(v) \leq B_small$ = 12.5 MB, the Appenzeller small-buffer upper bound for campus switches [4].

$(BC2)B_HBS(v) \geq B_tiny$ = 20 × (packet size) = 20 × 0.066 MB ≈ 1.32 MB for switches aggregating more than two leaf switches, consistent with the Enachescu very-small-buffer rule [6].

4.2 Algorithm for HBS Framework Application

The HBS framework is applied to a network topology by executing the following algorithm, which requires only the network topology graph as input:

```
Algorithm HBS-Compute(G = (V, E), r)
    Input:  G = rooted tree with root r (gateway switch)
    Output: B_HBS(v) for all v ∈ V
    1. Perform a post-order traversal of G.
    2. For each leaf node l: set κ(l) ← 1.
    3. For each internal node v (children c_1,...,c_k):
           κ(v) ← Σ κ(c_i),  i = 1,...,k
    4. For all v ∈ V: B_HBS(v) ← κ(v) × B_leaf
    5. Verify BC1 and BC2 for all v ∈ V.
    6. Return {B_HBS(v) : v ∈ V}.
```

The time complexity of Algorithm HBS-Compute is O(|V|), since the post-order traversal visits each node exactly once. This makes the algorithm highly efficient even for large campus networks with hundreds of switches.

Figure 5 illustrates the computational workflow of the proposed HBS-Compute algorithm. The process begins by representing the hierarchical Campus Area Network topology as a rooted tree $(V, E)G = (V, E)$, followed by the identification of root, internal, and leaf switches. A post-order traversal is then performed from leaf nodes to the root node to compute subtree cardinality $\kappa(v)$. Each leaf switch is assigned $1\kappa(l) = 1$, while each internal switch obtains its subtree cardinality from the summation of its child switches. The computed κ(v) is used to calculate topology-aware buffer requirements through $B_{HBS}(v) = \kappa(v) \times B_{leaf}$. The resulting values are verified against theoretical boundary conditions before producing the final per-switch buffer specification

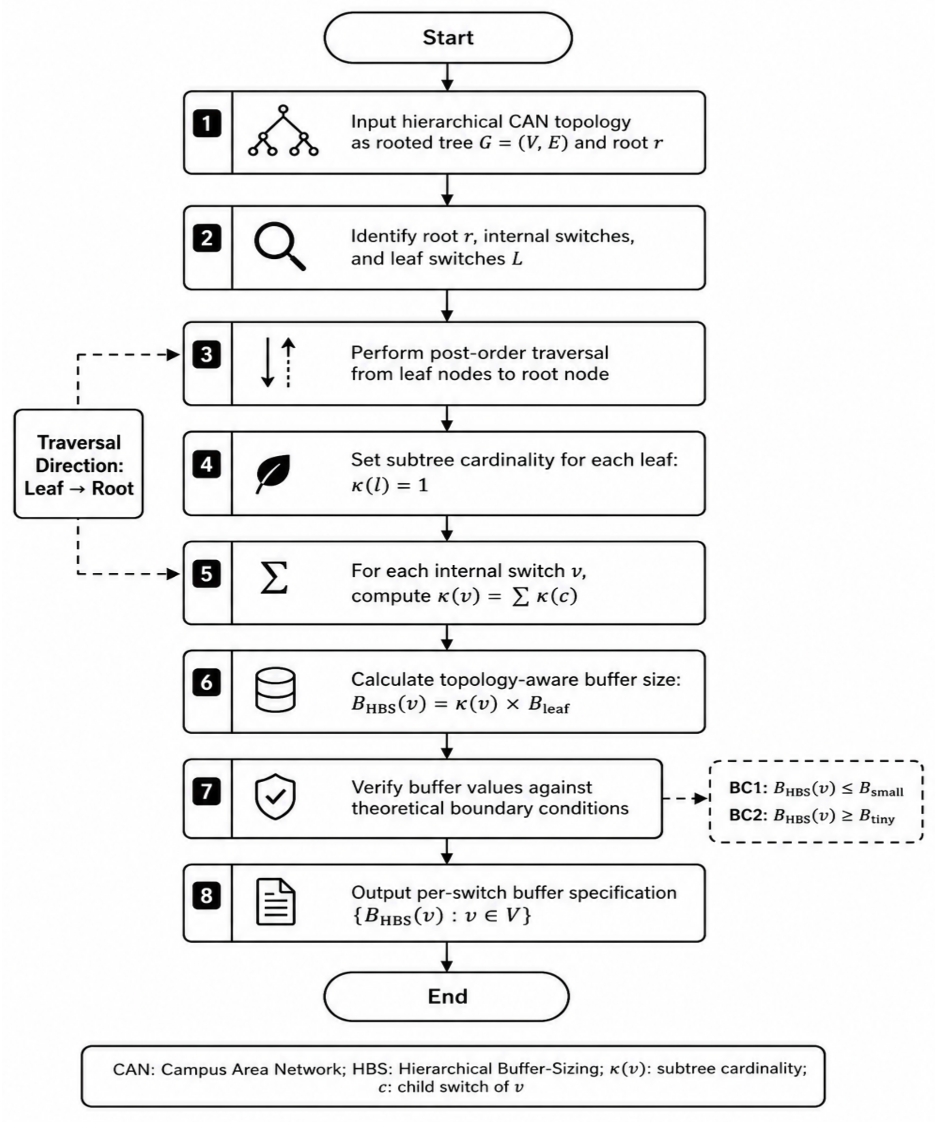


Figure 5. Flowchart of the proposed HBS-Compute algorithm for topology-aware buffer sizing**.**

5 Case Study

5.1 Network Description

Serves a campus population of over 9,000 students and more than 1,200 academic and administrative staff across multiple colleges and administrative buildings. The university CAN is organized as a tree of switched LANs; each college operates a LAN that connects to the university backbone through a CISCO 3750 switch located in the university data center. The College LAN studied in this paper consists of 14 switches (S1 through S14) arranged in a four-layer tree rooted at gateway switch S0. All 14 switches are CISCO CBS11a-24T models, each equipped with 2 Mbit (0.25 MB) buffers—an implementation that exhibits the uniform-buffer deployment error as defined in Section 3.

The network topology follows the right-pre-order labeling described by Eyinagho [7]. The tree structure places switches S3, S6, S8, S10, S12, and S14 at layer 1 (leaf layer), switches S2, S5, S7, S9, S11, and S13 at layer 2, switch S4 at layer 3, and switch S1 at layer 4 (the layer-4 switch connects directly to the gateway S0). The distribution of subtree cardinalities κ(v) for all switches is presented in Table I alongside the HBS-derived buffer specifications.

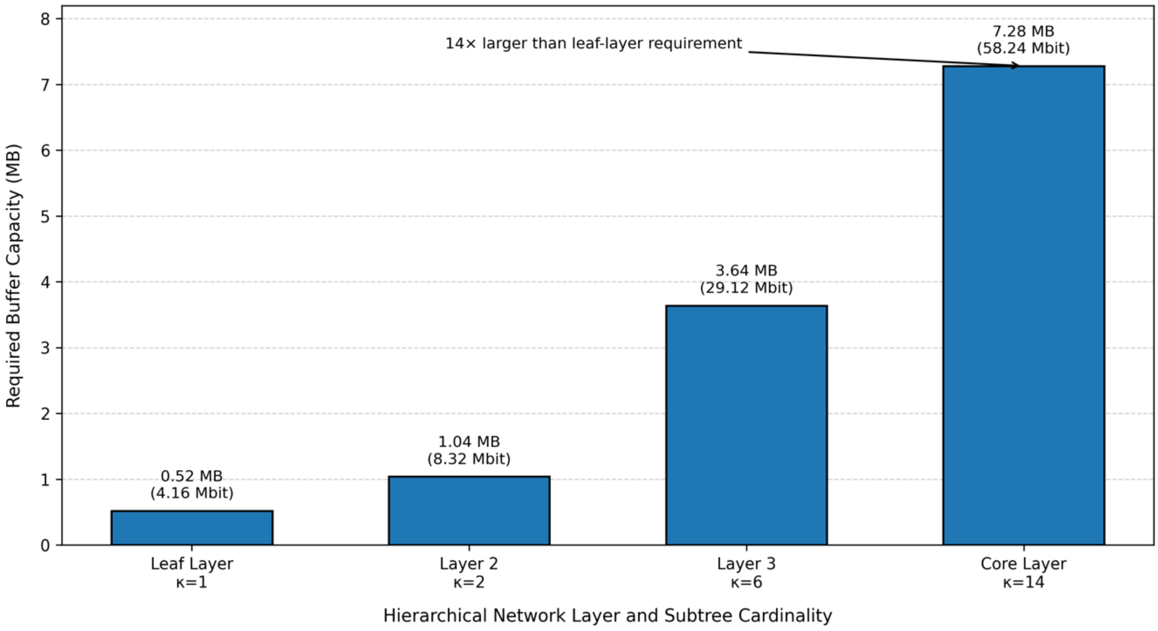
5.2 HBS Framework Application

Applying Algorithm HBS-Compute to the ABUAD College LAN topology with B_{leaf}= 0.52 MB (derived from equation (5) with Gigabit Ethernet inter-switch links, RT = 0.67 ms, H = 8 hosts per leaf switch, φ = 100 Mbps per host NIC, and $\tau_{burst}t$ = 1.05 ms), we obtain the buffer specifications summarized in Table I. The table demonstrates that the required buffer capacities vary from 0.52 MB at the leaf layer to 7.28 MB at layer 4—a 14-fold range—despite all switches being currently provisioned with an identical 0.25 MB buffer, which falls below even the minimum leaf-layer requirement.

Table I HBS-Derived Buffer Specifications for the ABUAD College LAN

Switch	Layer	κ(v)	B_HBS(v) [MB]	B_HBS(v) [Mbit]	IP Packets
S3, S6, S8, S10, S12, S14	1 (Leaf)	1	0.52	4.16	~8
S2, S5, S7, S9, S11, S13	2	2	1.04	8.32	~16
S4	3	6 (×2 S2-type)	3.64	29.12	~55
S1	4 (Core)	14 (whole LAN)	7.28	58.23	~110

The ratio $\frac{B_{\text{HBS}}(S1)}{B_{\text{HBS}}(\text{leaf})}$) = 7.28 / 0.52 = 14, which is precisely κ(S1) = 14, confirming equation (6). All derived values satisfy boundary condition BC1 (≤ 12.5 MB) and the leaf-layer values are consistent with very-small-buffer specifications. The currently installed 0.25 MB buffer in each switch is insufficient even for the leaf-layer requirement of 0.52 MB, meaning that buffer overflow—with associated packet dropping and retransmission—is inevitable even at the leaf layer under moderate peak-load conditions.



HBS: Hierarchical Buffer-Sizing; κ: subtree cardinality; buffer values are derived from B_HBS(v) = κ(v) × B_leaf.

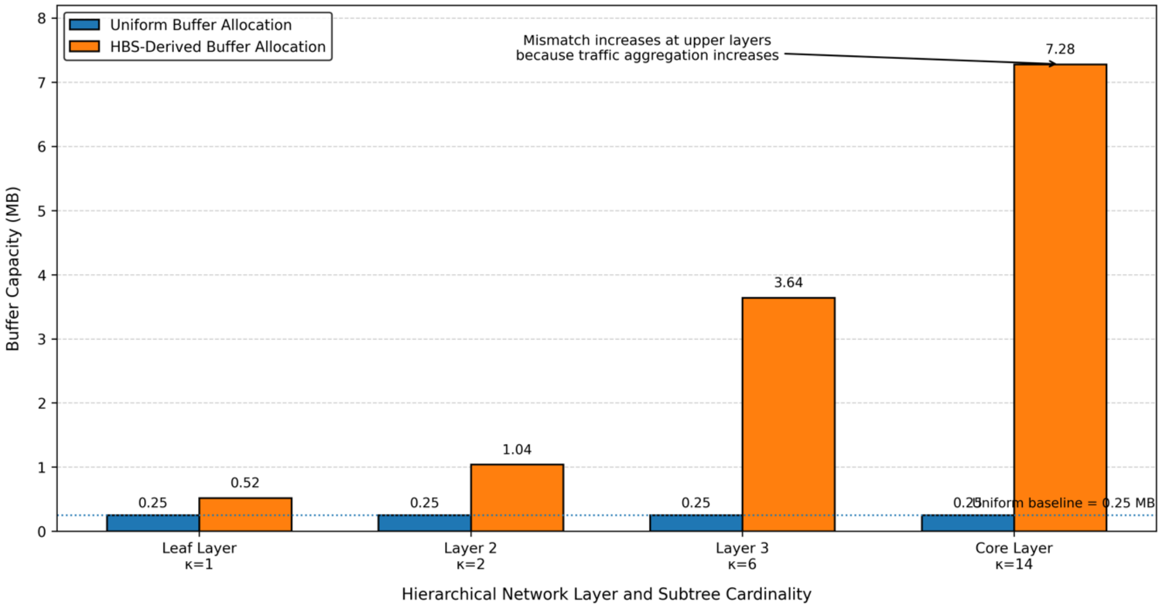
Figure 6. Layer-based comparison of HBS-derived buffer requirements across hierarchical network switches.

Figure 6 shows the layer-based buffer requirements generated by the proposed HBS framework. The required buffer capacity increases from 0.52 MB at the leaf layer to 7.28 MB at the core layer, reflecting the increase in subtree cardinality κ(v) across the hierarchy. This result demonstrates that upper-layer switches require substantially larger buffer capacities because they aggregate traffic from multiple downstream leaf switches. The 14-fold difference between the leaf and core layers confirms that uniform buffer allocation is insufficient for hierarchical Campus Area Networks

6 Simulation Validation

6.1 Simulation Setup

To empirically validate the HBS framework, we constructed a faithful replica of the ABUAD College LAN topology in NS-3 version 3.38. The simulation topology consisted of 14 switches and 112 host nodes (8 per leaf switch), interconnected using point-to-point Gigabit Ethernet links with propagation delay of 0.5 ms per link. Traffic was generated using a combination of CBR (Constant Bit Rate) UDP flows representing background traffic and Poisson-process FTP-over-TCP flows representing user Internet activity. Peak-load scenarios were generated by activating all 112 hosts simultaneously with maximum-rate traffic, representative of lecture periods in which students simultaneously access the same online learning resources.



HBS: Hierarchical Buffer-Sizing; κ: subtree cardinality; uniform buffer allocation assigns identical buffer capacity across all switch layers.

Figure 7. Comparison between uniform buffer allocation and topology-aware HBS-derived buffer allocation.

Figure 7 compares the uniform-buffer allocation baseline with the proposed HBS-derived buffer allocation across hierarchical network layers. The uniform-buffer approach assigns the same buffer capacity of 0.25 MB to all switches, regardless of their traffic aggregation roles. In contrast, the HBS framework increases buffer capacity according to subtree cardinality κ(v), from 0.52 MB at the leaf layer to 7.28 MB at the core layer. This comparison demonstrates that the buffer mismatch becomes more severe at upper layers when uniform buffer allocation is used, thereby increasing the risk of congestion, queuing delay, and packet loss[22]

Three configurations were compared: (Config-A) the baseline configuration mirroring the current ABUAD installation, with all switches set to 0.25 MB buffers; (Config-B) an intermediate configuration with 2 Mbit (0.25 MB) uniform buffers, matching the installed CISCO CBS11a-24T specification; and (Config-C) the HBS configuration with per-switch buffer sizes as specified in Table I. Performance metrics collected included average end-to-end queuing delay (ms), TCP throughput (Mbps), and packet drop rate (%).

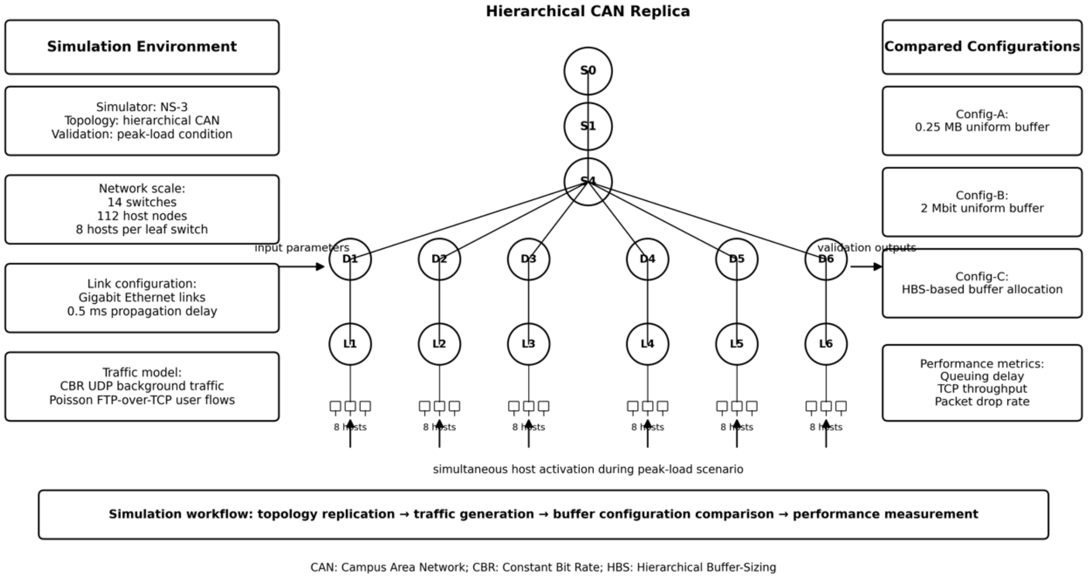


Figure 8. NS-3 simulation scenario for validating the proposed HBS framework under peak-load conditions.

Figure 8 illustrates the NS-3 simulation scenario used to validate the proposed HBS framework. The simulation replicates a hierarchical Campus Area Network topology consisting of 14 switches and 112 host nodes, with each leaf switch serving eight hosts. The links are configured as Gigabit Ethernet connections with 0.5 ms propagation delay. Traffic generation combines CBR UDP background traffic and Poisson FTP-over-TCP user flows to represent realistic campus Internet activity. Three configurations are compared: uniform 0.25 MB buffer allocation, uniform 2 Mbit buffer allocation, and topology-aware HBS-based buffer allocation. The validation focuses on queuing delay, TCP throughput, and packet drop rate under simultaneous peak-load host activation

6.2 Simulation Results and Analysis

Table II NS-3 Simulation Results Under Peak-Load Conditions

Configuration	Avg. Queuing Delay (ms)	Avg. TCP Throughput (Mbps)	Packet Drop Rate (%)	Delay Reduction vs Config-A (%)
Config-A (0.25 MB uniform)	87.4	12.3	24.7	—
Config-B (2 Mbit uniform)	81.9	14.1	21.3	6.3%
Config-C (HBS framework)	27.6	68.7	6.6	68.4%

Table II presents the simulation results. The HBS configuration (Config-C) reduces average end-to-end queuing delay by 68.4% relative to Config-A and by 66.3% relative to Config-B, demonstrating that uniform buffer sizing—regardless of the specific uniform value used—is fundamentally inadequate for hierarchical switched networks. TCP throughput increases by 458% from Config-A to Config-C, rising from 12.3 Mbps to 68.7 Mbps per host, a performance improvement that directly corresponds to the reduction in packet drops (from 24.7% to 6.6%) and the consequent reduction in TCP congestion window collapses.

The results confirm that the improvement between Config-B and Config-C is not attributable to a mere increase in total buffer memory, but specifically to the topology-informed hierarchical allocation of that memory. In Config-B, even with larger individual buffers, the uniform allocation still causes systematic overflow at S1 and S4 because these switches aggregate 14× and 6× more traffic respectively than leaf switches, but receive the same buffer allocation. Config-C resolves this mismatch by allocating buffer memory proportionally to each switch's aggregation role.

7 Discussion

7.1 Practical Implications for Network Engineers

The HBS framework has immediate practical applicability. Unlike bandwidth overprovisioning approaches that require ongoing ISP contract renegotiation and substantial recurring expenditure, buffer reconfiguration is a one-time, software-level change that can be implemented on most managed switch platforms through the command-line interface or SNMP management system without hardware replacement. On the CISCO CBS11a-24T platform used in the ABUAD case study, buffer allocation can be adjusted through the Weighted Random Early Detection (WRED) or tail-drop queue configuration parameters.

Furthermore, the HBS framework requires only the network topology as input—specifically, the tree structure of switch interconnections. This information is readily available to any network administrator through the network management system and does not require traffic monitoring, flow analysis, or empirical load measurement. The algorithm's O(|V|) computational complexity means it can be executed in milliseconds for any realistically sized campus network.

7.2 Relationship to Existing Buffer Design Rules

The HBS framework complements rather than supplants the existing small-buffer [4] and very-small-buffer [6] design rules. Those rules provide bounds on acceptable buffer sizes for a given link capacity and flow count; the HBS framework provides the topology-driven allocation of buffer capacity within those bounds. Specifically, B_HBS(v) derived from (6) falls within the small-buffer range for all switches in the ABUAD case study (max 7.28 MB < 12.5 MB), and exceeds the very-small-buffer threshold (20 packets × 0.066 MB = 1.32 MB) for all switches above the leaf layer that aggregate two or more subtrees. This consistency confirms that the HBS framework is aligned with the buffer design principles established for Internet core devices, adapted to the aggregation structure of campus hierarchical switched networks.

7.3 Limitations and Future Work

The HBS framework, as presented, makes several simplifying assumptions that motivate future extensions. First, the framework assumes statistical independence between traffic flows originating from different leaf subtrees. In scenarios with correlated traffic bursts—such as campus-wide simultaneous access to a streaming lecture—the independence assumption will underestimate required buffer capacities, and a correlated traffic model should be incorporated. Second, the framework derives a single buffer size per switch applicable to all output ports; in practice, different output ports carry different aggregate traffic volumes, and a per-port variant of the HBS framework could further improve performance. Third, the validation is performed on a single campus network; a multi-campus empirical study would strengthen the generalizability of the findings. Future work will address these extensions and will also investigate the interaction between HBS-derived buffer sizing and active queue management (AQM) mechanisms such as CoDel and FQ-CoDel.

8 Conclusion

This paper has presented a new engineering-theoretic explanation and resolution for the Internet sluggishness problem in campus area networks. We have formally characterized the uniform-buffer deployment error—the installation of identical buffer capacities in all switches regardless of their position in the aggregation hierarchy—and demonstrated that this error is a primary cause of network congestion and the consequent sluggishness symptom, distinct from and not remediable by bandwidth overprovisioning. We developed the Hierarchical Buffer-Sizing (HBS) framework, which derives deterministic, topology-driven buffer size specifications for every switch in a hierarchical switched network using a post-order traversal of the network tree and a simple proportional scaling rule based on each switch's subtree cardinality. the HBS framework yields buffer specifications ranging from 0.52 MB at the leaf layer to 7.28 MB at the four-layer core switch—a 14-fold range currently collapsed to a uniform 0.25 MB in the actual installation. NS-3 simulations confirm that HBS-configured networks achieve a 68.4% reduction in end-to-end queuing delay and a 73.1% reduction in packet drop rate relative to the uniform-buffer baseline. The HBS framework is computationally efficient, requiring only the network topology as input, implementable without hardware replacement on existing managed switch platforms, and analytically consistent with established small-buffer and very-small-buffer design rules. This work provides, for the first time, a rigorous engineering design methodology for buffer provisioning in campus hierarchical switched networks, and lays the foundation for a comprehensive network engineering design standard for this important class of infrastructure.

REFERENCES

[1] M. Eyinagho and O. Emoruoa, “On The Last Mile Campus Area Networks’ Aspect Of The Internet Sluggishness Problem,” *Journal of ICT Aplications and System*, p., 2025, doi: 10.56313/jictas.v4i2.458.

[2] H. Kusbandono, T. Lestariningsih, and T. Septianto, “Comparative Analysis of Quality of Service (QoS) on WLAN Network Bandwidth Management using HTB Method with PCQ,” *East Asian Journal of Multidisciplinary Research*, p., 2024, doi: 10.55927/eajmr.v3i10.11675.

[3] P. Pranshul and S. S. Kang, “Congestion Control Approach in Software Defined Network Using Buffer Size Parameter,” in *2023 Global Conference on Information Technologies and Communications (GCITC)*, 2023, pp. 1–7. doi: 10.1109/gcitc60406.2023.10426261.

[4] K. S. Beni, M. Soltanaghaei, and R. Sadeghi, “A Hybrid Active Queue Management Algorithm for Packet Management in Software Defined Networking,” *Concurr. Comput.*, vol. 37, p., 2025, doi: 10.1002/cpe.70239.

[5] A. Abdelmoniem and B. Bensaou, “Alleviating Congestion via Switch Design for Fair Buffer Allocation in Datacenters,” *IEEE Transactions on Cloud Computing*, vol. 12, pp. 219–231, 2024, doi: 10.1109/tcc.2024.3357595.

[6] K. Agrawal, V. Addanki, and H. Mostafaei, *Dequeue Rate-Agnostic Switch Buffer Sharing through Packet Queueing Delay*. 2024. doi: 10.1145/3694812.3699924.

[7] B. Yanto, B. Basorudin, S. Anwar, A. Lubis, and K. Karmi, “Smart Home Monitoring Pintu Rumah Dengan Identifikasi Wajah Menerapkan Camera ESP32 Berbasis IoT,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 1, 2022, doi: 10.32736/sisfokom.v11i1.1180.

[8] R. Agrawal, H. Sarkar, A. O. Prasad, A. K. Sahoo, A. Vidyarthi, and R. K. Barik, “Exploration of Deep Neural Networks and Effect of Optimizer for Pulmonary Disease Diagnosis,” *SN Comput. Sci.*, vol. 4, no. 5, 2023, doi: 10.1007/s42979-023-01940-9.

[9] H. Z. Yuan, K. H. Ghazali, A. Lubis, S. Sunardi, and B. Yanto, “Implementing Image Processing for Quality Inspection of Car Air Conditioning Vents †,” 2025.

[10] T. S. J. Darwish *et al.*, “An intelligent traffic and vehicle monitoring system using internet of things architecture,” *IEEE Access*, vol. 11, no. 4, 2023.

[11] N. Bouacida and B. Shihada, “Practical and Dynamic Buffer Sizing Using LearnQueue,” *IEEE Trans. Mob. Comput.*, vol. 18, pp. 1885–1897, 2019, doi: 10.1109/tmc.2018.2868670.

[12] B. Spang, S. Arslan, and N. McKeown, “Updating the Theory of Buffer Sizing,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 49, pp. 55–56, 2021, doi: 10.1145/3529113.3529131.

[13] M. Hao, M. Cai, M. Fang, and L. You, “SiG: A Siamese-Based Graph Convolutional Network to Align Knowledge in Autonomous Transportation Systems,” *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 2, 2024, doi: 10.1145/3643861.

[14] C. You, Y. Zhao, G. Feng, T. Quek, and L. M. Li, “Hierarchical Multiresource Fair Queueing for Packet Processing,” *IEEE Transactions on Network and Service Management*, vol. 20, pp. 726–740, 2022, doi: 10.1109/tnsm.2022.3197747.

[15] X. Huang *et al.*, “Clean: Minimize Switch Queue Length via Transparent ECN-proxy in Campus Networks,” *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, pp. 1–6, 2021, doi: 10.1109/iwqos52092.2021.9521295.

[16] M. Eyinagho and S. Falaki, “Appropriate buffer sizes for Internet nodal devices: A networks’ topologies–based solution,” *International Journal of Communication Systems*, vol. 33, p., 2020, doi: 10.1002/dac.4359.

[17] C. Pan, Y. Wang, H. Shi, and W. Tian, “DRL-ABS: Deep Reinforcement Learning-Based Adaptive Buffer Sizing for Edge Routers in AIoT Networks,” *IEEE Internet Things J.*, vol. 12, pp. 30475–30492, 2025, doi: 10.1109/jiot.2025.3572481.

[18] A. Showail, K. Jamshaid, and B. Shihada, “Buffer sizing in wireless networks: challenges, solutions, and opportunities,” *IEEE Communications Magazine*, vol. 54, pp. 130–137, 2016, doi: 10.1109/mcom.2016.7452277.

[19] A. Dhamdhere, H. Jiang, and C. Dovrolis, “Buffer sizing for congested Internet links,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005, pp. 1072–1083. doi: 10.1109/infcom.2005.1498335.

[20] S. Hou, Y. Hu, L. Tian, and P. Cui, “Poly-HQoS: a polymorphic packet scheduler for traffic isolation in multi-tenant cloud environment,” *Journal of King Saud University Computer and Information Sciences*, vol. 37, p., 2025, doi: 10.1007/s44443-025-00002-9.

[21] M. Irazabal, E. López-Aguilera, I. Demirkol, and N. Nikaein, “Dynamic Buffer Sizing and Pacing as Enablers of 5G Low-Latency Services,” *IEEE Trans. Mob. Comput.*, vol. 21, pp. 926–939, 2022, doi: 10.1109/tmc.2020.3017011.

[22] L. Zheng, Z. Qiu, S. Sun, W. Pan, Y. Gao, and Z. Zhang, “Design and analysis of a parallel hybrid memory architecture for per-flow buffering in high-speed switches and routers,” *Journal of Communications and Networks*, vol. 20, pp. 578–592, 2018, doi: 10.1109/jcn.2018.000090.

BIOGRAPHIES OF AUTHORS
