telecom

MDPI

Article

Automated Multivendor Network Topology Discovery and Shadow IT Detection Using Asynchronous Hybrid Polling and Graph-Based Anomaly Analysis

Jamil Qalandarov [1], Ahram Nishanov [1], Saidkul Saparov [2,3], Gulomjon Juraev [4], Ilhom Jumayev [5], Mengliev Shaydulla [6], Dinara Parpibaeva [7], Nargiza Salimova [7], Janibek Kurbanov [8] and Rashid Nasimov [2,8,*]

1 Department of Systems and Applied Programming, Tashkent University of Information Technologies named After Muhammad al-Khwarizmi, Tashkent 100084, Uzbekistan; djamic@gmail.com (J.Q.); admin@tuit.uz (A.N.)
2 Department of Artificial Intelligence, Tashkent State University of Economics, Tashkent 100200, Uzbekistan; s.saparov@tsue.uz
3 Department of Computer Engineering, Gachon University Sujeong-Gu, Seongnam-si 13120, Gyeong-gi-Do, Republic of Korea
4 Department of Information Technology, Mathematics and Engineering Systems International Innovative University, Qarshi City 180100, Uzbekistan; g.p.jurayev@mail.ru
5 Department of Information Technologies, Denau Institute of Entrepreneurship and Pedagogy, Denov 190500, Uzbekistan; i.jumayev@dtpi.uz
6 Department of Applied Mathematics, Termez State University, Termez 190100, Uzbekistan; shaydullamengliyev81@gmail.com
7 Department of Clinical Modeling, Tashkent State Medical University, Tashkent 100109, Uzbekistan; d.parpibaeva@tashmeduni.uz (D.P.); n.salimova@tashmeduni.uz (N.S.)
8 Department of Radio-Electronic Devices and Systems, Tashkent State Transport University, Tashkent 100167, Uzbekistan
* Correspondence: rashid.nasimov@tsue.uz

Abstract

The rapid growth of corporate networks and the wide use of "Bring Your Own Device" (BYOD) policies have made these networks harder to secure and manage. One visible consequence is the spread of unauthorized, unmanaged switches and wireless routers. Network Management Systems (NMSs) differ in their polling, topology discovery, and multivendor capabilities. This paper presents an automated framework for multivendor network topology discovery and Shadow IT detection, together with a topological mapping architecture based on concurrent routines written in the Go programming language. The contribution lies in the integration and operational evaluation of established techniques on a heterogeneous campus network rather than in new individual algorithms. A hybrid asynchronous scanning algorithm combining the Internet Control Message Protocol (ICMP), Simple Network Management Protocol (SNMP), Link Layer Discovery Protocol (LLDP), and Address Resolution Protocol (ARP) is used to overlap discovery-stage response waits. We use graph-theoretic models and a heuristic "Health Score" function to classify network anomalies such as loopbacks, Spanning Tree Protocol (STP) conflicts, and unauthorized devices. Virtualization-prefix and cached ESXi-signature checks exclude candidate ports associated with virtualization; Time-To-Live (TTL) values separately estimate the operating system (OS) family. Experiments conducted on a live university campus network showed that the 100-worker asynchronous pool achieved a 94.9-fold discovery-stage speedup over sequential polling. In a replay of a recent forwarding database (FDB) snapshot, the virtualization checks reduced the number of candidate ports with multiple media access control (MAC) addresses from 96 to 72. Detection accuracy against independent labels remains to be evaluated. The framework was further evaluated through a fixed 24-h monitoring window and a functional comparison with established network management

Check for updates

Academic Editor: Barbara M. Masini

Received: 22 September 2026
Revised: 23 October 2026
Accepted: 26 October 2026
Published: 2 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

platforms. The results demonstrate faster discovery on the evaluated campus network; detection accuracy, resource overhead, and generalization require further validation.

Keywords: shadow IT; network security; network topology discovery; anomaly detection; OS fingerprinting; asynchronous polling; SNMP; graph theory

1. Introduction

As information and communication technologies are deployed more widely, enterprise local area networks have become larger and more complex. Industry forecasts describe growth in the number of connected endpoints, driven by the proliferation of Internet of Things (IoT) devices, cloud-hybrid workloads, and institutional adoption of BYOD policies [1]. In practice, BYOD frameworks grant end users a degree of network access that, without adequate policy enforcement, enables the arbitrary attachment of personal consumer-grade routers and unmanaged Layer 2 switches to production infrastructure segments.

In the security literature, such unregulated hardware is usually described as Shadow IT, and it poses a different kind of risk from conventional endpoint threats [2–4]. In this study, Shadow IT refers specifically to unauthorized network infrastructure devices, such as consumer-grade switches and wireless routers, that are physically connected to an institutional network without being represented in the authorized network inventory. Unlike rogue software or unauthorized cloud services, unmanaged switches can forward frames at Layer 2 while absent from management inventories. They may neither expose SNMP Management Information Bases (MIBs) nor advertise their neighbors via LLDP [5–7]. An unmanaged device is not necessarily unauthorized. Campus administrators reported both permitted small hubs and unauthorized Wi-Fi routers causing unintended Dynamic Host Configuration Protocol (DHCP) responses; these are local observations, not prevalence estimates. Over 181 days of retained monitoring history, the system recorded 354 distinct switch-port candidate identifiers, of which 52 recurred on at least 170 separate dates. These are repeated candidate observations rather than confirmed unauthorized devices, but they show that access ports with multiple MAC addresses are a persistent rather than an exceptional phenomenon in the studied network. Rogue DHCP servers in particular are a recognized operational threat for which countermeasures at the switch level have been standardized [8]. An unmanaged switch inserted into an access-layer segment can extend an existing broadcast domain, trigger unintended STP topology changes, and, if loop prevention is not configured, cause a broadcast storm that affects the whole network [9,10]. We use the term topological blindness for the situation in which the administrator's logical topology map no longer matches the physical network. This term reflects the long-recognized importance of accurate physical topology knowledge for network management [11]. This mismatch between the administratively known topology and the physically deployed topology creates a security visibility problem, since an unauthorized Layer 2 device may remain operational while being invisible to conventional inventories based on Internet Protocol (IP) addressing and SNMP.

Several approaches address network visibility and anomaly detection, but each has limitations when applied to Shadow IT. Software-Defined Networking (SDN) separates the control plane from the data plane through protocols such as OpenFlow and can improve visibility within its deployment scope [12]. However, migration can involve substantial equipment and operational costs. Consequently, hybrid deployments in which legacy switches coexist with SDN components remain common [13], and network monitoring in

such environments still depends largely on data collected from existing devices through SNMP polling and LLDP neighbor discovery [13]. Therefore, a security-oriented discovery mechanism that operates over existing heterogeneous infrastructure, without requiring replacement of deployed network hardware, remains practically relevant for campus and public-sector networks.

Within classical NMS tooling, commercially available platforms such as Cisco Prime Infrastructure and SolarWinds Network Topology Mapper (NTM) provide automated Layer 2/Layer 3 discovery and topology visualization. Managing heterogeneous multivendor networks remains an active automation problem [14]. Recent work also addresses device hardening in multivendor environments [15]. In contrast, mature open-source platforms such as Zabbix and OpenNMS provide reliable metric collection and threshold alerting [16], and recent research has proposed methods for discovering physical connection relationships between devices from collected network data [17]. Commercial and open-source management tools support different combinations of discovery, monitoring, visualization, and integration. Their device-discovery and topology features should not be conflated with the specific task of identifying wired multi-MAC candidate ports from FDB information. The documentation-based comparison in Section 3.5 states the reviewed scope and marks capabilities that could not be established from the selected sources. It does not assert universal absence of FDB-based security extensions in other platforms.

A third body of work addresses network anomaly detection through intrusion detection systems based on machine learning. While these approaches demonstrate strong performance on traffic classification tasks, they often operate on packet or flow features; physical-port localization depends on the available infrastructure context. This capability is essential for the actionable remediation of Shadow IT hardware [18–22]. Therefore, anomaly detection at the traffic level and topology discovery at the infrastructure level address different layers of the security problem. This distinction motivates an approach that explicitly associates detected anomalies with specific network devices, ports, and Layer 2 segments.

1.1. Practical Objectives

The implementation addresses four practical objectives. First, it reduces the latency that sequential or partially concurrent polling can introduce when large address spaces contain inactive hosts. Second, it integrates Layer 2/Layer 3 observations across heterogeneous vendors. Third, it suppresses the false-positive Shadow IT alerts that FDB-based detection can produce in virtualized environments, where bridged hypervisors may expose multiple MAC addresses behind a single physical port. Fourth, it combines retained anomaly suggestions into a single device score through the proposed network health metric. These objectives do not imply that existing NMS products lack these capabilities.

To address the four objectives identified above, this paper presents an automated framework for multivendor network inventory, observed topology reconstruction, and Shadow IT detection. The system consists of concurrent routines written in Go and uses Go's goroutine scheduler to run the scanning pipeline asynchronously and in parallel. A hybrid discovery algorithm combining ICMP reachability probes, SNMPv2c MIB traversal, LLDP neighbor table retrieval, and ARP cache inspection integrates management observations across heterogeneous access-layer environments. To suppress false-positive Shadow IT alerts arising from virtualization infrastructure, we apply virtualization-prefix and cached HTTPS ESXi-signature checks. IEEE vendor lookup and IP TTL heuristics separately enrich device and OS-family labels. Candidate anomalies, including loopback self-references, STP conflicts, and suspected unauthorized device attachment, are modeled using graph-theoretic constructs applied to the reconstructed topology graph. A Health

Score function combines the anomaly indicators of each device into a single normalized value to support administrator review; ranking utility was not independently validated. Real-time alerting is delivered via a Telegram bot integration, and inventory reports are generated automatically in DOCX format. Telemetry is stored continuously in an InfluxDB time-series database, where it can later be used for predictive analysis.

The main contributions of this paper, which concern the integration and operational evaluation of established techniques rather than fundamentally new individual algorithms, are as follows:

1. An asynchronous multiprotocol network discovery architecture implemented in Go that integrates ICMP, SNMP, LLDP, and ARP for multivendor topology reconstruction.
2. A topology-aware Shadow IT detection mechanism that combines FDB analysis with virtualization-prefix and cached ESXi-signature checks to exclude candidate ports associated with virtualization hosts.
3. A graph-based anomaly classification framework and composite Health Score model for reviewing topology, stability, and security-related suggestions.
4. An empirical evaluation on a heterogeneous university campus network comprising four/24 subnets (1016 usable host addresses) and 68 successfully scanned managed switches, including worker-count timing, candidate-port analysis, continuous monitoring, and a documentation-based functional NMS comparison.

Based on these objectives, the study addresses the following research questions:

RQ1: Does asynchronous multiprotocol polling reduce ICMP/basic-SNMP discovery time compared with sequential polling in a heterogeneous campus network?

RQ2: How do virtualization-prefix and cached ESXi-signature checks reduce FDB-based candidate-port outputs in the available snapshot?

RQ3: Can the proposed composite Health Score reproduce the saved device scores, and how sensitive is its ranking to the penalty weights?

The study hypothesizes that asynchronous multiprotocol polling will substantially reduce discovery-stage time, while virtualization checks will reduce candidate-port outputs associated with virtualized hosts. Independent accuracy and administrator-priority validation remain outside the available evidence.

The remainder of this paper is organized as follows. Section 2 presents the mathematical models, anomaly detection logic, and system architecture underlying the proposed framework. First, the network topology is formalized using graph theory (Section 2.1). Next, the Shadow IT detection algorithm and virtualization discrimination mechanism are described (Section 2.2). The heuristic Health Score function that aggregates all detected anomalies into a single metric is then defined (Section 2.3). Finally, the system architecture and parallel scanning strategy are outlined (Section 2.4). Section 3 reports four evaluations (parallel scanning performance, candidate-port filtering, Health Score distribution, and 24-h monitoring observations) and a documentation-based comparison with existing NMS platforms. Section 4 discusses the implications of the findings, contextualizes the results against related work, and identifies the limitations of the current system. Section 5 concludes the paper and outlines directions for future research.

1.2. Related Work

Early automated topology discovery systems relied on SNMP MIB-II traversal to enumerate Layer 2 and Layer 3 connectivity [11,23,24]. While effective in homogeneous environments, single-protocol approaches degrade in heterogeneous multivendor infrastructures where SNMP community strings are inconsistently configured. Hybrid algorithms that combine SNMP, LLDP, and ICMP were later proposed and were shown to discover more devices than single-protocol methods [25,26]. Their performance depends on protocol

coverage, timeouts, and concurrency settings. Go is well suited to concurrent network services, although concurrency introduces its own implementation pitfalls [27,28]. In this study, the effectiveness of goroutine-based concurrency for large-scale multiprotocol NMS polling in heterogeneous campus networks is examined for the discovery stage on one campus network.

Gartner predicts that by 2027, 75% of employees will acquire, modify, or create technology outside IT's visibility, up from 41% in 2022 [29], and shadow IT has been found to take especially diverse forms in higher education institutions [30]. Rogue switch detection has been studied previously [31]. FDB analysis has been proposed as a passive rogue switch detection mechanism, based on the observation that multiple MAC addresses on a single non-trunk port can indicate downstream equipment. Open-source tools such as NeDi and Netdisco support network discovery [32,33]. FDB-based analysis is conceptually related to rogue access point detection [34]. However, this heuristic can produce false positives in virtualized environments, where bridged virtual machines present FDB signatures identical to those of unauthorized physical switches. The UK National Cyber Security Centre discusses broader asset management for Shadow IT [35]. This study examines an explicit sequence of port and virtualization checks in the implemented workflow.

OS fingerprinting research demonstrates that IP TTL values and IEEE Organizationally Unique Identifier (OUI) MAC prefixes can provide lightweight but imperfect clues about device type and OS family [36,37]. Related work on IoT device-type identification has further shown that lightweight network-level features can support device classification [38,39]. In this work, TTL is used only to estimate the OS family; it does not control the Shadow IT filter.

Graph-theoretic analysis, including breadth-first search (BFS) [40], provides a natural basis for detecting structural irregularities such as excessive hop distances in access-layer topologies. Gujral et al. proposed a composite network health score that combines security vulnerability indicators into a single value that administrators can use for prioritization [41]. This idea resembles severity-weighted vulnerability scoring schemes such as the Common Vulnerability Scoring System (CVSS) [42]. The present heuristic summarizes retained device suggestions; a score alone does not establish prioritization accuracy.

Finally, time-series database systems, including InfluxDB, have been benchmarked specifically for monitoring workloads that combine continuous ingestion with analytical queries [43]. InfluxDB was adopted in this work as the storage layer for real-time telemetry.

Recent work also examines generative AI in network management [44]. Miuccio et al. [45] study multi-agent reinforcement learning for wireless medium-access protocols. Their learned policies address protocol behavior, whereas this system combines existing SNMP/LLDP observations and predefined detection rules. Such learning-based approaches assume control over the communication protocol itself, whereas the present work operates on legacy equipment from multiple vendors whose management protocols cannot be modified. Moreover, because LLDP neighbor tables and FDB tables are themselves retrieved via SNMP, combining these sources does not remove the dependence on correct SNMP access, and ICMP reachability cannot recover inaccessible management tables. Learning adaptive polling priorities or intervals from the monitoring history is a possible extension, but it has not been implemented or evaluated here.

2. Materials and Methods

2.1. Network Topology Modeling Based on Graph Theory

To analyze the network topology, the observed network is modeled as a directed graph with string-valued edge labels:

$$\begin{aligned} G_t &= (V_t, E_t^s, \ell_t), \quad V_t \subset \Sigma^*, \\ E_t^s &\subseteq V_t \times V_t, \quad \ell_t : E_t^s \to \Sigma^*. \end{aligned} \tag{1}$$

Here Σ^* is the set of finite strings. V_t contains stored node IDs, including device, endpoint, inferred hub, and hypervisor IDs; E_t^s contains ordered adjacency pairs; and ℓ_t assigns their stored string labels, which may be empty. Labels may contain port, virtual local area network (VLAN), or rate text; they are not numerical traversal weights. Stored adjacency is not a verified physical-port multigraph. Edge insertion order can affect direction, and repeated observations between the same IDs can collapse to a single edge.

Based on this model, candidate topological anomalies can be identified mathematically. A Loopback candidate is a self-reference in a device record d and reported port p, not proof of a physical broadcast storm. Let $L_{d,p}$ and $F_{d,p}$ be its LLDP and FDB record sequences:

$$B_t(d,p) = \Big[\exists l \in L_{d,p} : b_L(d,l)\Big] \vee \Big[\exists f \in F_{d,p} : b_F(d,f)\Big]. \tag{2}$$

In practice, the Loopback condition is verified via the LLDP neighbor table and MAC address table. Let N normalize a MAC string, $c_d = N(\text{ChassisID}_d)$, ι_t indicate a configured ignored issue, and τ_d indicate a trunk port:

$$\begin{aligned} b_L(d,l) &= [N(l.\text{RemChassisId}) = c_d] \wedge \neg \iota_t(i_L(d,l)), \\ b_F(d,f) &= [N(f.\text{MACAddress}) = c_d] \\ &\quad \wedge \neg \tau_d(f.\text{PortNumber}) \wedge \neg \iota_t(i_F(d,f)). \end{aligned} \tag{3}$$

Here i_L and i_F are the issue keys of the respective records, and p is the port index. If a switch detects its own MAC address under these conditions, the system registers a Loopback anomaly. This does not detect every multi-switch cycle. Physical-port interpretation also depends on interface mapping, VLAN, STP, and link aggregation.

A BFS algorithm is used to flag excessive stored-graph depth. The configured root r is traversed along outgoing stored edges at unit cost:

$$\begin{aligned} d_{t,r}(v) &= \min_{P: r \rightsquigarrow v} |E(P)|, \quad v \in R_{t,r}, \\ \mathcal{A}_{t,h}(r) &= \{v \in R_{t,r} : d_{t,r}(v) > h\}, \quad h_* = 3. \end{aligned} \tag{4}$$

Here $R_{t,r}$ is the set of nodes reachable from r, P is a directed path, and $|E(P)|$ is its edge count. The threshold used here is h* = 3 hops. Any node exceeding this threshold is flagged for architecture review; the threshold is not a measured latency boundary or authorization rule. In the saved 48-node aggregate, thresholds 1, 2, 3, 4, and 5 flag 9, 5, 0, 0, and 0 nodes, respectively.

The congestion state of each link was intended to be assessed using the Edge Utilization Ratio:

$$\begin{aligned} U_{\kappa,j} &= \frac{r_j^{\text{in}} + r_j^{\text{out}}}{c_{\kappa,j}}, \quad c_{\kappa,j} > 0, \\ C_\kappa^{\text{intended}} &= \mathbf{1}\{\exists j \in J_\kappa : U_{\kappa,j} \geq 0.80\}. \end{aligned} \tag{5}$$

For interface key κ and supplied samples J_κ, r_{in} and r_{out} are bit rates and c is the positive interface capacity; 1{condition} is an indicator. The intended threshold is 80% of capacity. The inspected code instead uses a fixed 1 Gbps denominator and triggers above 800 Mbps;

this implementation error has not been corrected or re-evaluated in this study. Interface telemetry is keyed separately from graph edges; missing telemetry is not zero load.

While the above models address topological and traffic-level anomalies, campus networks also face security threats from unauthorized devices that are invisible to standard discovery protocols. The following subsection formalizes the candidate-detection logic for such Shadow IT segments.

2.2. Detection Logic and Virtualization Discrimination

Unmanaged switches that form Shadow IT segments may lack management IP addresses and LLDP support. To identify candidate ports, logical conditions are applied to the switch's FDB. For switch a and port p, F is the FDB record sequence and F^{nr} is its non-randomized subset; repeated VLAN records are not deduplicated. The Shadow IT detection condition is formalized as:

$$\begin{aligned} S_{t,a,p} = \mathbf{1}\Big\{\Big|\mathcal{F}^{nr}_{a,p}\Big| > 1\Big\}\,\mathbf{1}\{\neg\tau_{a,p}\}\,\mathbf{1}\{\neg\iota_t(i_S(a,p))\} \\ \times\mathbf{1}\{\phi_{a,p} \le 0.5\}\,\mathbf{1}\{\nexists f \in \mathcal{F}_{a,p} : N(f.\text{MACAddress}) \in K_t\}. \end{aligned} \tag{6}$$

However, a single port may host multiple MAC addresses because legitimate virtual machines (VMware, Hyper-V, KVM) operate in bridged mode. To minimize false positives (FPs), the checks run in this order: more than one non-randomized MAC record; no trunk flag τ; no configured ignore key i_S; virtualization-prefix fraction $\varphi \le 0.5$; and no FDB MAC in the cached HTTPS ESXi set K_t. The denominator of φ is all FDB records on that port. If more than half of the records match the configured virtualization prefixes, the port is excluded; a cached ESXi match also excludes it. The IEEE OUI database [46] and the IP packet TTL separately enrich vendor and Linux/Windows OS-family labels. TTL is not used as a virtual-machine exclusion test. Beyond static port-MAC analysis, the system also monitors dynamic interface behavior. Interface stability is assessed by counting the supplied status records whose NewStatus value equals 2:

$$n^{\text{down}}_{\kappa} = \sum_{j \in M_{\kappa}} \mathbf{1}\{\text{NewStatus}_j = 2\}, \quad C^{\text{stability}}_{\kappa} = \mathbf{1}\{n^{\text{down}}_{\kappa} > 0\}. \tag{7}$$

Here M_κ indexes the supplied status records for interface κ. Any positive count creates a stability suggestion; it is not a verified count of physical flaps. The inspected export path did not populate the Events input needed to validate this rule historically.

The anomaly indicators defined in Sections 2.1 and 2.2 are aggregated into a single composite metric, as described in the following subsection.

2.3. Heuristic Health Score Function

To quantify the retained anomaly burden of each device, a heuristic "Health Score" function was formulated. Operating on a 100-point scale, the function subtracts penalty weights based on the count of each retained suggestion type:

$$\begin{aligned} \mathcal{H}_q(\boldsymbol{\pi}) &= \max\left(0, 100 - \sum_{i \in \mathcal{I}} \pi_i x_{i,q}\right), \quad \pi^{(0)}_i = 10, \\ \overline{\mathcal{H}}^{(0)} &= \frac{1}{|\mathcal{Q}|}\sum_{q \in \mathcal{Q}} \mathcal{H}_q(\boldsymbol{\pi}^{(0)}). \end{aligned} \tag{8}$$

Here q is a device row in the export Q, $x_{i,q}$ counts retained suggestions of type i assigned to that row by exact IP match, and π_i is the penalty per entry.

The historical device export uses $\pi_i = 10$ for every type. This weighting differs from that used in the snapshot and network-summary scoring paths. The complete set of

anomaly types, their severity labels, and the corresponding penalty weights is summarized in Table 1. The mean is the arithmetic mean of the 68 device scores.

Table 1. Historical device score penalty weights by anomaly type.

Anomaly Type	Severity	Penalty Weight (π_i)
Loopback self-reference	Critical	10 per entry
STP Root Mismatch	High	10 per entry
Shadow IT	High	10 per entry
Congestion	High/Medium	10 per entry
Port Flapping	High	10 per entry
Physical Errors	Medium	10 per entry
Topological Depth > 3 hops	Medium	10 per entry

The penalty weights were assigned as a ten-point baseline per retained suggestion, rather than a calibrated severity scale. The saved export contains 103 Shadow IT suggestions and two loopback suggestions on one device; the other types are absent from this export, not necessarily from the network. These weights should therefore be interpreted as heuristic rather than universally optimal parameters. Section 3.3 reports sensitivity to the two observed suggestion types.

2.4. System Architecture and Asynchronous Hybrid Discovery

As illustrated in Figure 1, the automated inventory system uses a concurrent application architecture implemented in the Go (Golang) programming language. Sequential SNMP polling spends a large amount of time waiting for responses from inactive IP addresses; to avoid this delay, an asynchronous "Worker Pool" strategy is used. A hybrid mechanism first issues rapid ICMP Echo-Requests across the Classless Inter-Domain Routing (CIDR) block. Only active nodes ($V_{\text{snmp}} \subseteq V_{\text{icmp}}$) are subsequently passed to the SNMP Data Collection module for basic SNMP discovery. Detailed MIB-II, LLDP-MIB, and Q-BRIDGE-MIB collection follows [5,7,47,48]. The processed data are normalized, converted into a JavaScript Object Notation (JSON) object, and streamed to an InfluxDB Time-Series Database. Anomalies trigger real-time callbacks via the Telegram Bot API.

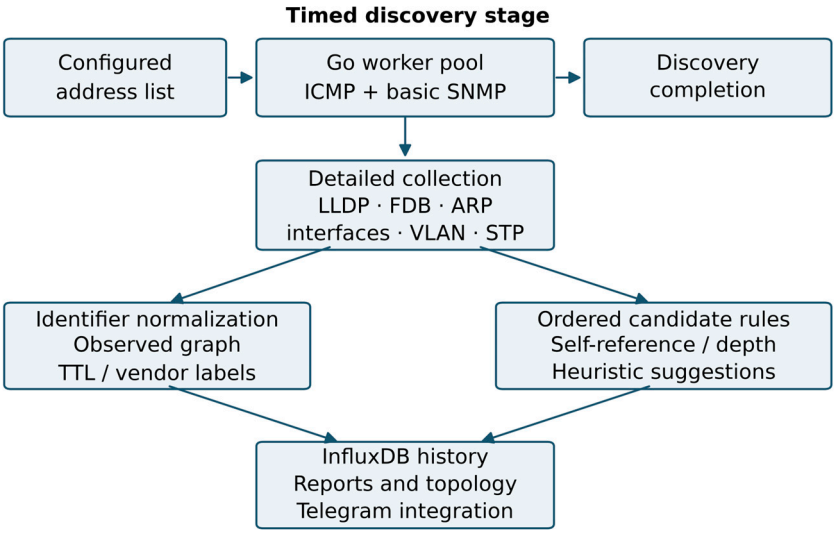


Figure 1. System architecture and data flow pipeline; only initial ICMP/basic-SNMP discovery is timed.

The parallel scanning speedup coefficient is defined as:

$$S(W) = \frac{T(1)}{T(W)}, \tag{9}$$

where T(1) is the sequential execution time and T(W) is the execution time with W parallel workers. These times cover ICMP/basic-SNMP discovery, including worker setup and waiting, but exclude detailed collection, topology construction, and export. The parallel processing efficiency is:

$$E(W) = \frac{S(W)}{W} \times 100\%. \tag{10}$$

The current deployment uses SNMPv2c and shared community credentials on multiple switches. This is a deployment choice; separate credential management and migration to SNMPv3 are planned. Telegram management uses the configured group ID and assigned personnel. This arrangement does not establish complete command authorization, secure credential storage, least-privilege access, or management-plane isolation; those controls require a deployment security audit. Until SNMPv3 is deployed, the recommended mitigations are read-only community strings, SNMP access control lists that restrict queries to the collector address, and confinement of management traffic to a dedicated management VLAN. Whether these measures are applied on every switch in the current deployment has not yet been audited.

3. Results

The system was assessed through four evaluations: parallel scanning performance (Section 3.1), candidate-port filtering (Section 3.2), Health Score distribution analysis (Section 3.3), and 24-h monitoring observations (Section 3.4). Section 3.5 provides a documentation-based comparative review of existing NMS platforms.

3.1. Testbed Configuration and Polling Performance

The proposed framework was evaluated on the live production campus network of the Tashkent University of Information Technologies (TUIT) with official institutional authorization from the university's IT department. The monitoring was limited to network-management metadata and topology information required by the proposed framework; application payloads and user content were not inspected. The scanning configuration was also designed to minimize operational impact on the production network. The experiment was conducted without interrupting ongoing academic and administrative activities. The original March 2026 evaluation reported 1017 IP addresses distributed across four major subnets: 172.20.254.0/24, 172.20.253.0/24, 172.20.230.0/24, and 172.20.240.0/24. The historical target manifest was not recovered. The later timing campaigns used 1016 unique host addresses from these four/24 ranges, excluding network and broadcast addresses. The historical inventory reported 69 managed switches in total, of which 68 active managed switches were successfully discovered and scanned: 42 SNR switches (comprising SNR-S2985G-24TC, SNR-S2985G-48T, and SNR-S2995G-24FX models running SNR-OS), 23 HPE switches (including HP V1910-48G, HP V1910-24G, HPE V1910-24G, HP A5120-48G SI, and HPE Comware Platform Version 7.1.070, Release 3506/3506P02), 2 Huawei switches (S5700 series), and 1 switch from an unknown vendor. The remaining switch failed SNMP scanning but was mapped via LLDP link topology. The network was configured with 22 active VLANs (including VLAN IDs 1, 10, 11, 12, 26, 45, 160, 164, 168, 1216, 1414, 1500, 1501, 1612, 1616, 1620, 1624, 1632, 1636, 1992, 3333, and 3334). A collapsed two-tier topology design was deployed, with access-layer switches connected via 10 Gbps uplinks (core layer) and 1 Gbps downlinks (access layer). The BFS depth analysis from the main HPE

core switch (Chassis ID 4c:ae:a3:27:6d:8a) at IP address 172.20.254.1 identified 48 reachable graph nodes: 1 at Depth 0, 38 (79.17%) at Depth 1, 4 at Depth 2, and 5 at Depth 3. The exact historical graph and per-node types were not retained, so these 48 nodes are not the complete 68-switch inventory. The wireless network layer consisted of 37 Ubiquiti Wi-Fi access points (APs) (including U6-Pro, U6-LR, U7-LR, UAP-AC-LR, and UAP-AC-Mesh-Pro models). STP was enabled on all 68 scanned switches (100%), with Multiple Spanning Tree Protocol (MSTP) as the primary protocol and Rapid Spanning Tree Protocol (RSTP) as the fallback. The original evaluation associated 3682 unique MAC addresses with these 68 managed switches, 37 Wi-Fi APs, 116 VMware ESXi virtual machines (bridged mode) running on 6 physical hypervisor hosts, and 2316 user endpoints (PCs, laptops, printers, IP cameras, VoIP phones, and BYOD smartphones, with approximately 1200 active BYOD devices observed during peak hours).

The original March 2026 benchmark used a scanning server (host) equipped with an Intel Xeon E-2334 CPU (4 physical cores, 8 threads, 3.40 GHz base frequency), 16 GB of DDR4 RAM, and a 1 Gbps uplink network interface card (NIC). The host ran Ubuntu 22.04 LTS (Linux kernel 5.15) and was physically connected directly to the core HPE switch at IP address 172.20.254.1 to minimize latency. During a concurrent scan with a worker pool of W = 100, the scanner CPU utilization peaked at 4.2%, RAM consumption remained under 38 MB, and peak network bandwidth consumption was 2.4 Mbps, as originally reported. Raw measurements supporting these historical hardware and resource-use figures were not recovered, so low overhead is not established.

The software stack for the mapping system was compiled using Go 1.25.0 and used the gosnmp v1.42.1 library for SNMP transactions, influxdb-client-go/v2 v2.14.0 for writing metric streams to an InfluxDB v2.7.6 time-series database (configured with an infinite retention policy), pyTelegramBotAPI v4.31.0 and Flask v3.1.3 (Python 3.10) for the Telegram Bot API alerting service, and python-docx v1.2.0 for automated report generation. The system loaded the IEEE OUI vendor database [46] (MA-L, MA-M and MA-S registries, last updated on 20 September 2024) for device classification. The protocol configuration included an ICMP ping timeout of 1.0 s with 0 retries (1 packet only), an SNMP discovery timeout of 2.0 s with 1 retry (for initial ping checks), and an SNMP walk timeout of 5.0 s with 1 retry (for bulk walking tables via snmpbulkwalk with MaxRepetitions set to 10). These values describe the configuration used for continuous monitoring. In both the original benchmark and the October 2026 campaigns, the discovery stage used a 2 s ping limit. Because most addresses in the target ranges are inactive, this limit dominates the sequential time (about 1.87 s per address), which explains the close agreement of the W = 1 times (1904.1 s and 1906.3 s). During continuous monitoring, known switches were re-polled every 5 min, while interface metrics were collected approximately every 30 s. A monitor snapshot-and-analysis loop and a separate analysis routine each ran approximately every 60 s. A monitor cycle is one loop execution, not a complete address-space scan. These configured intervals are distinct from the logged execution counts in Section 3.4.

To evaluate the efficiency of the asynchronous Worker Pool, the scanning time was measured as a function of the number of concurrent workers (W). As shown in Table 2, increasing the worker pool to 100 yielded a 94.9× speedup. The ICMP/basic-SNMP discovery stage was completed in 20.1 s, compared with 1904.1 s using a single worker. The parallel efficiency at W = 100 reached E(100) = 94.9% within this worker-count range. The reduction in scanning time with increasing worker pool size is illustrated in Figure 2.

Table 2. (A) Historical discovery-stage time and speedup S(W) (one retained observation per worker count). (B) Additional discovery timings in seconds ($n = 4$; sample standard deviation (SD) uses $n - 1$).

(A)			
Workers (*W*)	**Scan Time (s)**	**Speedup *S*(*W*)**	**Efficiency *E*(*W*)**
1	1904.1	1.0×	100.0%
10	190.5	10.0×	100.0%
25	76.3	25.0×	99.8%
50	38.3	49.7×	99.4%
100	20.1	94.9×	94.9%
(B)			
Workers W	**Mean**	**Median**	**Sample SD**
1	1906.327354	1905.704635	1.439587
10	192.328635	192.330529	0.008431
25	78.074943	78.095953	0.060975
50	40.053839	40.050966	0.006493
100	20.091839	20.085276	0.022882

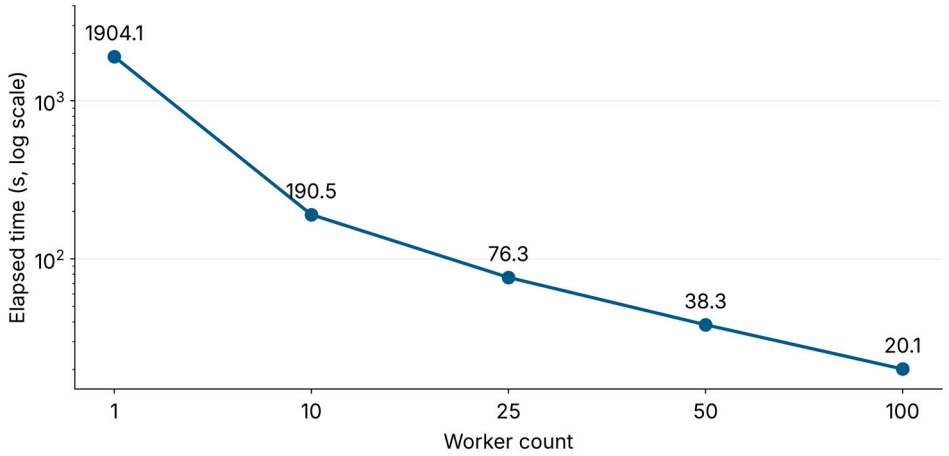


Figure 2. Historical discovery-stage time reduction as a function of the parallel worker pool size.

Four additional campaigns conducted on 16 October 2026 provided n = 4 runs per worker count (20 scans in total). These campaigns used a host with two Intel Xeon Silver 4210R CPUs, 20 physical cores, 40 logical CPUs, approximately 94.044 GiB of RAM, and Ubuntu 24.04.3 LTS. A Go 1.25 harness used the unchanged scanner with SNMPv2c over User Datagram Protocol (UDP) port 161, a 3 s SNMP timeout, one retry, and a 2 s ping limit. Runs followed W = 1, 10, 25, 50, 100 with 30 s pauses. Production monitoring was stopped and restored around each campaign; other services remained running. These are later runs, not repeated observations of the original benchmark.

Paired 100-worker speedups were 95.081479, 94.690038, 94.860242, and 94.891465. SNMP successes at W = 50 were 66/69/65/68 and at W = 100 were 67/69/70/68; equal totals did not always contain the same responding devices. The fixed run order and the single target-set size limit the interpretation of these results. Scalability with increasing network size and resource overhead were not measured in these campaigns.

3.2. Virtualization Filtering and Shadow IT Detection

The reference set used to evaluate Shadow IT detection was reported to consist of 40 ports identified as hosting unmanaged switches, with 38 detections and two misses. The contemporaneous port-by-port administrator records were not recovered. Table 3

retains the reported numbers, but independent labels and a complete confusion matrix remain unverified.

Table 3. Historical reported detection figures; independent reference labels were not recovered.

Detection Mode	False Positives	Precision	Recall	F1-Score	Reported Shadow IT Detected
Reported filter DISABLED	149	20.3%	95.0%	33.5%	38 ports
Reported filter ENABLED	4	90.5%	95.0%	92.7%	38 ports
FP Reduction	97.3%	—	—	—	—

The historical evaluation reported a reduction in FP alerts during Shadow IT detection. As shown in Table 3, enabling the filter reduced the number of FP instances from 149 ports to 4 ports, representing a 97.3% FP reduction. As a result, detection precision rose from 20.3% to 90.5%, while the overall F1-Score improved from 33.5% to 92.7%. The reported Recall was 95.0%, based on 38 detections out of the 40-port reference set. The causes of the two misses and four residual false positives could not be reconstructed. TTL does not control the candidate filter, and these improvements are not attributed to it.

A separate replay of the 15 October 2026 snapshot produced 222 multi-MAC switch–port pairs. The private-MAC check removed none; trunk exclusion removed 126, configured ignore removed none, virtualization-prefix majority removed 3, and cached ESXi matching removed 21, leaving 72. With the other stages fixed, applying neither virtualization rule, prefixes only, HTTPS only, and both rules left 96, 93, 72, and 72 candidates, respectively. This unlabeled rule comparison is not an accuracy ablation or a replacement for the historical 40-port reference set. The 96 candidates that remain without virtualization rules in this snapshot are not directly comparable with the 149 false positives reported historically with the filter disabled, because the two results come from different dates and the historical trunk handling could not be reconstructed.

The authors confirmed that four examined switch-to-switch links were authorized and that the Ignore option had been applied to those ports during testing. This does not independently identify the historical four false positives. A verified physical testbed map, complete port labels, and device provenance tied to that historical reference set remain unavailable.

3.3. Health Score Distribution

Using the retained device suggestions, the Health Score function (Equation 8) was applied to evaluate the overall condition of each switch. In the saved export of all 68 successfully scanned switches, the mean Health Score was 84.6 points. As shown in Table 4 and Figure 3, half of the devices (50.0%) scored in the excellent range (90–100), while 2 devices were in the critical score category.

Table 4. Health Score distribution across 68 switches.

Health Score Range	Device Count	Share
90–100 (Excellent)	34	50.0%
70–89 (Good)	29	42.6%
50–69 (Warning)	3	4.4%
0–49 (Critical)	2	2.9%

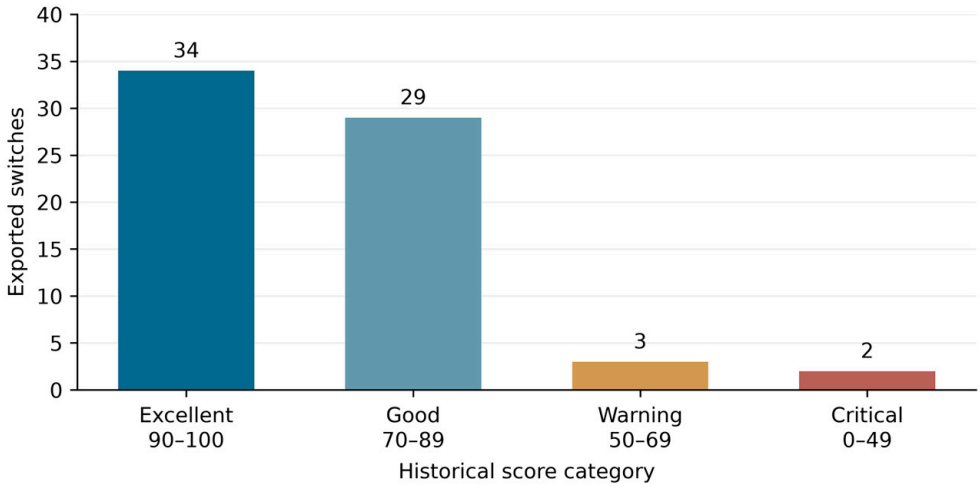


Figure 3. Switch distribution by Health Score quality ranges.

Recalculation with ten points per exported suggestion reproduced all 68 scores, a sum of 5750, and a mean of 84.558824 (84.6 rounded). Varying the Shadow IT and loopback penalties independently over {5, 8, 10, 12, 15} gave 25 scenarios. Spearman's rank correlation ranged from 0.987760 to 1.000000, and Kendall's τ_b ranged from 0.978645 to 1.000000; up to 13 strict rank inversions, a 24-place average-rank shift, and 28 category changes occurred. Uniform scaling preserved ranks but could change categories. The 11 baseline worst-set members, including boundary ties, remained in every scenario; one additional device could enter this set. No independent administrator ranking or sensitivity data for other suggestion types were available.

3.4. 24-h Monitoring Dynamics

While the above experiments assessed the system at a single point in time, a fixed 24-h window was reanalyzed to evaluate monitoring under operational conditions. Tables 5–7 and Figure 4 use 10 April 2026, 00:00–24:00 Coordinated Universal Time (UTC) (half-open interval [start, stop)), equivalent to 05:00–05:00 local time (UTC+05:00). Earlier summaries mixed the 4–5 April run with a 72-h export from 8–11 April and combined different counting units. The missing raw history of the earlier run cannot be reconstructed. In the selected window, device_events.mac records 30,830 JOIN and 33,132 LEAVE observations. Table 6 and Figure 4 use the same six equal four-hour aggregates. These are stored observations, not necessarily unique physical connection events.

Table 5. Monitoring totals for 10 April 2026, 00:00–24:00 UTC.

Parameter	Value
Monitor snapshot cycles	1450
Parallel analysis executions	1439
Completed known-switch re-poll batches	287
DEVICE_JOIN observations	30,830
DEVICE_LEAVE observations	33,132
Stored InfluxDB field values	15,884,005
Historical disk volume	Not verified

Table 6. Recorded device observations in equal four-hour UTC intervals on 10 April 2026.

Interval (UTC)	JOIN	LEAVE	Total
00:00–04:00	5198	5581	10,779
04:00–08:00	9036	7834	16,870
08:00–12:00	7822	7349	15,171
12:00–16:00	2976	6060	9036
16:00–20:00	3094	3528	6622
20:00–24:00	2704	2780	5484

Table 7. Stored points and field values in the same 24-h window.

Measurement	Points	Field Values
anomaly_summary	2889	17,334
device_events	63,962	191,886
interface_metrics	1,725,297	15,527,673
network_events	67,336	132,677
network_summary	2887	14,435
Total	1,862,371	15,884,005

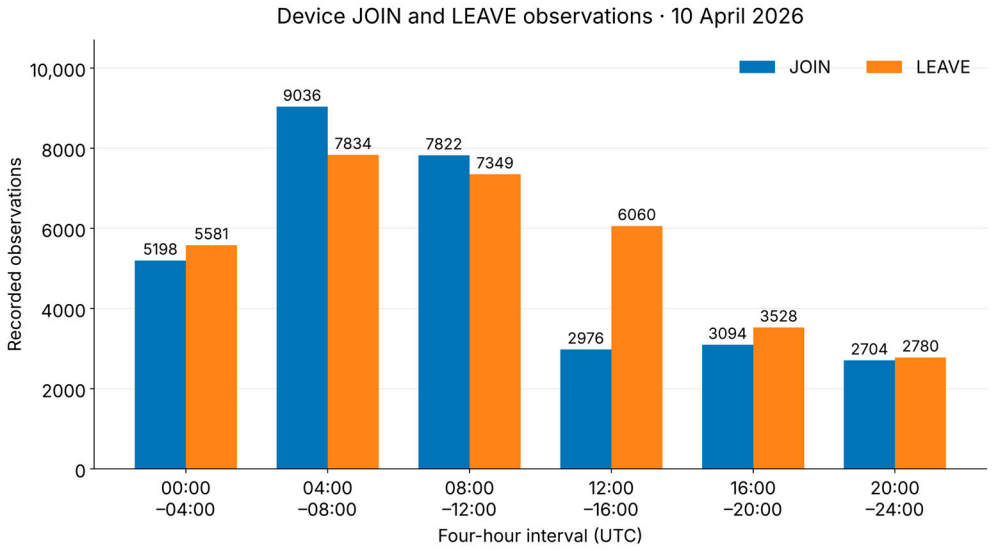


Figure 4. Distribution of device JOIN (blue) and LEAVE (orange) observations from Table 6 for 10 April 2026, 00:00–24:00 UTC.

The distribution of stored data across InfluxDB measurements is presented in Table 7. One point groups fields with the same measurement, tags, and timestamp; each field value is counted separately. The 15,884,005 field values exactly match the total in Table 5. health_score is a field of network_summary and is not counted as an additional measurement. Disk volume was not independently recovered. The log and the network_events measurement contain two more LEAVE observations than device_events; the write discrepancy remains unresolved. Separate analysis paths may record related observations, and the available identifiers do not support physical-event deduplication. A restart and interface-series gaps also preclude a claim of uninterrupted sampling of every interface.

3.5. Documentation-Based Functional Comparison

Table 8 provides a documentation-based functional comparison of the proposed prototype with Zabbix, Paessler PRTG Network Monitor, SolarWinds Network Topology Mapper, and Cisco Prime Infrastructure. The reviewed product versions or documentation

editions and access date are stated explicitly. Core discovery and analysis capabilities are separated from licensing and operational integrations. Documented functionality, partial scope, integration-based support, and capabilities not established from the reviewed sources are distinguished. Device discovery is not treated as proof of unauthorized-device detection, and reported operating-system or virtualization inventory is distinguished from active OS-family inference. The comparison does not report cross-product runtime or detection-accuracy experiments.

Table 8. (A) Documentation-based comparison of core capabilities. (B). Licensing and operational integrations.

(A)					
Capability	**Zabbix 7.0**	**PRTG Network Monitor**	**SolarWinds NTM 2.3**	**Cisco Prime Infrastructure 3.10**	**Proposed System**
A1. Automatic L2 link reconstruction	No built-in network discovery; extension claim NE [49]	NE; discovery and custom maps documented [50,51]	D [52]	P: protocol discovery and link map documented; complete L2 reconstruction coverage unestablished [53,54]	D in code: LLDP/FDB links; coverage depends on management data
A2. Automatic L3 relationship reconstruction	No built-in network discovery [49]	NE [50,51]	D: subnet/next-hop relationships [52]	NE for a separate routed L3 graph in reviewed sections [53,54]	P: gateway subnet discovery; routed L3 graph NE
A3. Multi-vendor monitoring/ discovery scope	D: named vendor templates [55]	D: vendor-specific/generic SNMP sensors [56]	D: documented vendor examples [52]	P: third-party SNMP monitoring with feature limits [57,58]	P: standard MIB code and limited campus evidence; per-model completeness unverified
A4. FDB-based wired multi-MAC port candidates	NE [49]	NE; new-device discovery is a different task [51]	NE [52]	NE for this general rule [59]	D in code: candidate ports with exclusions; authorization not independently established
A5. Wireless rogue AP detection and wired port tracing	NE [49]	NE [51]	NE [52]	D for controller-reported rogue APs/switch port tracing, subject to license and switch prerequisites [59]	NE for this wireless workflow
A6. Reported OS/software or virtualization inventory	D: agent OS information [60]	D: WMI/SNMP system information [61]	P: system and virtualization inventory [52]	D: managed software-image inventory [62]	P: labels from management information; general version inventory NE
A7. Active OS-family inference from network responses	NE [60]	NE [61]	NE [52]	NE [62]	P: TTL family labels and ESXi HTTPS markers; general accuracy unvalidated
A8. Aggregation of health/status across signals or objects	D: configurable service-status hierarchy [63]	D: Business Process multi-object aggregation [64]	NE [52]	P: health policies and named Site Health Score; formula/coverage unestablished [57,65]	D: heuristic penalty score; distinct network/device paths and validation limits
(B)					
Attribute	**Zabbix 7.0**	**PRTG Network Monitor**	**SolarWinds NTM 2.3**	**Cisco Prime Infrastructure 3.10**	**Proposed System**
B1. Licensing	Open source, AGPL-3.0 [66]	Commercial editions; 100-sensor freeware also documented [67]	Commercial licensed product [68,69]	Commercial licensed product; entitlements apply [58,70]	Open-source license NE in reviewed copies
B2. Telegram alerts	I-official: Telegram webhook [71]	I-official: documented HTTP action [72,73]	NE [52]	NE for Telegram-specific support [74]	I-project: Go caller plus Python 3.10 bot; current bot deployment unverified
B3. DOCX audit report export	NE; PDF dashboard reports documented [75]	NE; HTML/PDF/CSV/XML reports documented [76]	NE [52]	NE; CSV/PDF event reports documented [74]	I-project: Python conversion route and earlier converter; current completeness unverified, GOST compliance NE
B4. Interactive SSH/Telnet access from management UI	Embedded terminal NE; SSH monitoring checks are distinct [77]	Embedded terminal NE; SSH sensors are distinct [78]	I-external: Windows Telnet/custom tools; embedded browser terminal NE [52]	I-external: client/browser helpers; embedded terminal NE [79]	P: WebSocket SSH/Telnet backend; current frontend and runtime unverified

The comparison uses the maintained Zabbix 7.0 documentation and release/7.0 integration, the PRTG online manual accessed on 16 October 2026, the archived SolarWinds NTM 2.3 guide, and the Cisco Prime Infrastructure 3.10 guides (with referenced 3.x licensing). Installed historical builds were not established; adjacent product suites and separate topology extensions are excluded. The proposed system is represented by the source-code snapshot inspected on 16 October 2026, with earlier external helpers identified separately. D = documented/implemented; P = partial; NE = not established, not absent; No = explicitly unsupported. I-official, I-project, and I-external identify integration types. These are documentation-based capabilities, not the measured comparative accuracy or runtime.

Integration type and prerequisites are reported separately from core discovery and analysis capabilities. An external-client launch is not equivalent to an embedded browser terminal. A generated DOCX file is not evidence of compliance with a named standard.

The prototype uses a Go worker pool. Programming language is an implementation choice and is not included among the comparative capabilities. Zabbix 7.0 also documents asynchronous pollers. The prototype's sequential/parallel benchmark evaluates its own worker configurations and does not establish a performance advantage over the products listed here [66].

4. Discussion

4.1. Asynchronous Polling Architecture and Scalability

The results support the discovery-time hypothesis: the goroutine-based asynchronous polling architecture completed ICMP/basic-SNMP discovery much faster than the sequential polling configuration. The 94.9-fold speedup observed at W = 100 workers demonstrates near-linear scaling within the evaluated worker range, with an efficiency of 94.9% for this workload. Amdahl's Law [80] provides a general framework for serial-work limits, not a fitted explanation of these measurements. This matters because most of the scanning time is spent not on computation but on I/O waiting, mainly on ICMP and SNMP timeouts from inactive addresses. While one goroutine waits for a response, the Go scheduler switches to another one, so the waiting periods overlap instead of adding up, and no heavy OS-level threads are required. Historical resource-use figures in Section 3.1 remain unverified. The hybrid ICMP-first filtering strategy further contributes to this efficiency by eliminating unnecessary protocol overhead for addresses that do not respond to ICMP.

Although scaling remained nearly linear between W = 1 and W = 100, the result describes worker-count scaling on this address set, not increasing-network-size scalability. The bottleneck at higher worker counts was not measured. Running several instances in parallel could allow horizontal scaling, but this needs to be tested on much larger networks.

From an operational point of view, virtualization filtering is an important part of the proposed workflow. Without filtering, the historical MAC FDB-based Shadow IT evaluation reported 149 false-positive alerts. With this many false alarms, the detection mechanism would be of little practical use in virtualized environments. With filtering enabled, the false-positive count of 4 ports corresponded to a detection precision of 90.5% and an F1-Score of 92.7%, with a reported Recall of 95.0% (38 of 40 reference ports detected). The independent labels and error causes remain unverified, as explained in Section 3.2. The snapshot replay verifies candidate filtering, while TTL provides an OS-family label.

The causes of the four reported residual false positives remain unverified. As future work, extending the fingerprinting module with additional behavioral signals (ARP request frequency, DHCP hostname patterns, or multicast Domain Name System (mDNS) service advertisements) could further reduce the residual false-positive rate, subject to independent accuracy evaluation. We plan to incorporate these signals in future work.

The Health Score distribution shows that most of the 68 scanned switches had high scores, while a small number fell in the critical score category.

Both devices in the critical range ($H < 50$) had retained Shadow IT suggestions, which explains their low scores under the historical ten-point rule. This does not independently validate the penalty weights or confirm unauthorized devices. The sensitivity analysis shows that overall rank correlation can remain high while individual ranks and categories change. However, broader multi-site validation is required before these weights can be considered generally applicable. In addition, the Health Score summarizes retained suggestions across 68 switch management interfaces in a single ranked list; its prioritization utility still requires comparison with administrator judgments.

The 24-h monitoring window showed how recorded campus network activity changes during the day. Static fixed-interval polling allocates equal scanning resources during periods of minimal recorded activity (20:00–24:00 UTC, 5484 observations) and peak activity (04:00–08:00 UTC, 16,870 observations). These counts do not establish unique physical events or a causal link to academic working hours. An adaptive polling interval strategy could reduce the number of scan cycles during low-activity periods and increase scanning frequency during predicted peak windows. Its effect on storage and detection delay remains to be tested. Retained candidate-count history covers 181 UTC dates, with 1–10 October absent. The repeated presence of these candidates suggests that a portion of the detected attachments was persistent rather than transient; it does not establish that they were unauthorized. The monitoring capability can inform administrator review and remediation planning.

The comparison identifies differences in functional scope and integration rather than an overall ranking. The proposed prototype combines LLDP/FDB-based link construction, heuristic identification of wired multi-MAC candidate ports, and penalty-based anomaly summaries. None of these components is new in isolation. The contribution lies in combining them in one operational workflow and evaluating that workflow on a production campus network. These candidates require administrative verification before being treated as unauthorized equipment. The prototype's OS-related labels use limited heuristics and should not be equated with a validated general-purpose fingerprinting system.

The established products provide related functions with different scopes. Zabbix documents service-status aggregation and an official Telegram webhook, and PRTG documents multi-object Business Process aggregation and an HTTP-based Telegram configuration. Cisco Prime Infrastructure documents third-party monitoring and a rogue-AP workflow that can trace detected access points to wired switch ports, subject to prerequisites. These functions are not interchangeable with the prototype's wired candidate-port rule. The reviewed sources do not establish that any product has lower accuracy on the campus network used in this study [57–59,63,64,71–73].

The prototype's operational extensions include a Python-based Telegram service, a DOCX conversion pipeline, and a WebSocket terminal backend. These features address deployment needs but do not establish an algorithmic advantage. Compliance with a specific GOST standard has not been verified, and an open-source licensing claim requires a corresponding project license. The availability of source code and the choice of Go are not sufficient evidence of that license.

4.2. Limitations and Future Directions

Several limitations constrain the current system's applicability and suggest directions for future research. First, the accuracy of directed graph construction depends fundamentally on the quality of SNMP and LLDP responses. The testbed revealed significant protocol coverage disparities across vendors. One Huawei S5700 switch was discoverable only

via LLDP link inference from its neighbors, with no direct SNMP data. These coverage gaps introduce potential blind spots in the topology graph that could cause Shadow IT devices connected downstream of non-responsive switches to evade detection entirely. Second, the current system operates exclusively in an observational and alerting capacity. Upon detecting a candidate Shadow IT port, the system generates a Telegram notification and updates the Health Score but takes no automated remediation action. Adding SDN/OpenFlow-based port shutdown would provide an enforcement mechanism whose operational effect would need evaluation. Finally, the vendor-labeling component uses a static OUI database last updated in September 2024. As new hardware manufacturers and virtual machine platform vendors emerge, the database requires periodic updates to maintain vendor coverage. Automating monthly synchronization of the OUI database with the IEEE MA-L registry would address this maintenance requirement.

Another limitation concerns the evaluation scope. The experimental validation was conducted on a single university campus network with a specific vendor composition, topology, traffic pattern, and address-space configuration. Although the testbed includes SNR, HPE, and Huawei equipment, the results may not generalize directly to enterprise, industrial, or geographically distributed networks with different vendor implementations and network architectures.

A further limitation concerns the establishment of the Shadow IT reference set. The reported precision and recall values depend on the completeness and correctness of the ground-truth labels. The historical port-level records were not recovered, so these metrics cannot presently be independently validated. Future evaluations need labels for alerted and non-alerted ports, a complete confusion matrix, and a verified physical testbed map.

5. Conclusions

This article presents a prototype system designed for inventory automation, observed topology reconstruction, and Shadow IT device detection in multivendor networks. Its contribution lies in the integration and operational evaluation of established techniques rather than in new individual algorithms. The system is implemented as concurrent routines written in the Go programming language. It also includes a hybrid discovery mechanism based on ICMP, SNMP, LLDP, and ARP, graph-based anomaly classification, and a composite Health Score function. The system was tested on an active university campus network to evaluate its performance. This network comprised four/24 subnets, 69 managed switches (68 successfully scanned), and approximately 3682 unique MAC addresses reported in the original evaluation. The experimental evaluation yielded four key findings. First, an asynchronous scanning pool based on 100 worker goroutines achieved a 94.9-fold speedup compared to the sequential polling method. The timed ICMP/basic-SNMP discovery stage took 20.1 s; complete topology construction and resource overhead were not established by this timing. Second, in the 15 October 2026 snapshot, the virtualization-prefix and cached ESXi checks reduced the candidate ports with multiple MAC addresses from 96 to 72. The historically reported 97.3% reduction in false positives could not be re-verified because the independent port labels were not recovered. Third, the Health Score function ranked 68 scanned switches using the historical ten-point penalty per retained suggestion. Both devices in the critical range carried multiple retained Shadow IT candidate suggestions. The score supports review, but administrator prioritization utility has not been independently validated. Fourth, the documentation-based comparison in Table 8 identifies differences in functional scope and operational integrations. It does not establish an overall ranking, cross-product runtime or accuracy superiority, or compliance with a specific GOST standard.

The main limitations of the current system are its reliance on the availability of SNMP and LLDP protocols and incomplete MIB coverage in certain HPE and Huawei firmware versions. Furthermore, the system operates strictly in an observational mode; it lacks port-level automated remediation capabilities.

Future research will focus on three main areas: implementing automated remediation mechanisms based on SDN/OpenFlow or NAPALM for verified Shadow IT ports; developing adaptive polling intervals tailored to observed temporal activity patterns; and extending the OS fingerprinting module with behavioral signals, such as DHCP hostnames and mDNS service advertisements, to further reduce the remaining false positives.

Author Contributions: Conceptualization, J.Q. and R.N.; methodology, J.Q. and A.N.; software, A.N.; validation, S.S. and G.J.; formal analysis, J.Q.; investigation, J.Q., D.P. and N.S.; resources, I.J.; data curation, M.S.; writing—original draft preparation, J.K.; writing—review and editing, R.N. and S.S.; visualization, A.N.; supervision, R.N.; project administration, J.Q.; funding acquisition, R.N. and J.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors received no external funding. The article processing charge (APC) was covered by the corresponding author.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The network monitoring data generated and analyzed in this study are not publicly available because they contain sensitive information about the institutional network infrastructure. The experimental methodology, system configuration, aggregated results, and evaluation metrics are provided in the article.

Acknowledgments: The authors acknowledge the academic and research environment and institutional support provided by Tashkent University of Information Technologies named after Muhammad al-Khwarizmi during the preparation of this study. During the preparation of this manuscript, the authors used Claude Opus 5.5 (Anthropic) to assist with language editing and improving the clarity of the text. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cisco Systems. *Cisco Annual Internet Report (2018–2023) White Paper*; Cisco: San Jose, CA, USA, 2020.
2. Haag, S.; Eckhardt, A. Shadow IT. *Bus. Inf. Syst. Eng.* **2017**, *59*, 469–473. [CrossRef]
3. Fürstenau, D.; Rothe, H.; Sandner, M. Leaving the shadow: A configurational approach to explain post-identification outcomes of shadow IT systems. *Bus. Inf. Syst. Eng.* **2021**, *63*, 97–111. [CrossRef]
4. Hense, V.; Westner, M.; Rösl, S.; Schieder, C. Charting the shadows: A bibliometric analysis of shadow IT. In *Business Modeling and Software Design (BMSD 2026)*; Shishkov, B., Ed.; Lecture Notes in Business Information Processing; Springer: Cham, Switzerland, 2026; Volume 588, pp. 99–114. [CrossRef]
5. McCloghrie, K.; Rose, M. *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II*; RFC 1213; IETF: Fremont, CA, USA, 1991. [CrossRef]
6. Presuhn, R. (Ed.) *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*; RFC 3416; IETF: Fremont, CA, USA, 2002. [CrossRef]
7. *IEEE Std 802.1AB-2016*; IEEE Standard for Local and Metropolitan Area Networks—Station and Media Access Control Connectivity Discovery. IEEE: Piscataway, NJ, USA, 2016.
8. Gont, F.; Liu, W.; Van de Velde, G. *DHCPv6 Shield: Protecting Against Rogue DHCPv6 Servers*; RFC 7610; IETF: Fremont, CA, USA, 2015. [CrossRef]
9. *IEEE Std 802.1Q-2022*; IEEE Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks. IEEE: Piscataway, NJ, USA, 2022.
10. Kurose, J.F.; Ross, K.W. *Computer Networking: A Top-Down Approach*, 8th ed.; Pearson: Hoboken, NJ, USA, 2021.

11. Breitbart, Y.; Garofalakis, M.; Jai, B.; Martin, C.; Rastogi, R.; Silberschatz, A. Topology discovery in heterogeneous IP networks: The NetInventory system. *IEEE/ACM Trans. Netw.* **2004**, *12*, 401–414. [CrossRef]
12. Kreutz, D.; Ramos, F.M.V.; Veríssimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined networking: A comprehensive survey. *Proc. IEEE* **2015**, *103*, 14–76. [CrossRef]
13. Khorsandroo, S.; Gallego Sánchez, A.; Tosun, A.S.; Arco, J.M.; Doriguzzi-Corin, R. Hybrid SDN evolution: A comprehensive survey of the state-of-the-art. *Comput. Netw.* **2021**, *192*, 107981. [CrossRef]
14. Alyahya, H.; Sheltami, T. NAS: AI-driven network automation for multi-vendor environments. *Arab. J. Sci. Eng.* **2026**, *51*, 12707–12719. [CrossRef]
15. Imoukhuede, A.B.; Sheltami, T.R.; Mahmoud, A.H.; Barnawi, A.Y. Optimization of network device hardening in a multivendor environment. *Sci. Rep.* **2025**, *15*, 15042. [CrossRef] [PubMed]
16. Velasco, A.R.H.; Malla, E.E.G.; Herrera, R.D.C.C.; Arévalo, F.D.M. Real-time monitoring and alerting system using Zabbix and Grafana software for wireless Internet access service management. In *Proceedings of the 2023 18th Iberian Conference on Information Systems and Technologies (CISTI)*; IEEE: New York, NY, USA, 2023; pp. 1–6. [CrossRef]
17. Wang, J.; Rui, J.; Niu, H.; Chang, Y.; Hu, J. Distributed physical device connection relationship discovery technology based on traffic information. In *Proceedings of the 7th International Conference on Information Science, Communication and Computing*; Springer: Singapore, 2023; pp. 28–41. [CrossRef]
18. Tightiz, L.; Nasimov, R.; Nasab, M.A. Implementing AI Solutions for Advanced Cyber-Attack Detection in Smart Grid. *Int. J. Energy Res.* **2024**, *2024*, 6969383. [CrossRef]
19. Ali, B.S.; Ullah, I.; Al Shloul, T.; Khan, I.A.; Khan, I.; Ghadi, Y.Y.; Abdusalomov, A.; Nasimov, R.; Ouahada, K.; Hamam, H. ICS-IDS: Application of big data analysis in AI-based intrusion detection systems to identify cyberattacks in ICS networks. *J. Supercomput.* **2024**, *80*, 7876–7905. [CrossRef]
20. Nascita, A.; Aceto, G.; Ciuonzo, D.; Montieri, A.; Persico, V.; Pescapé, A. A survey on explainable artificial intelligence for Internet traffic classification and prediction, and intrusion detection. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3165–3198. [CrossRef]
21. Safarov, F.; Basak, M.; Nasimov, R.; Abdusalomov, A.; Cho, Y.I. Explainable Lightweight Block Attention Module Framework for Network-Based IoT Attack Detection. *Future Internet* **2023**, *15*, 297. [CrossRef]
22. Abdulganiyu, O.H.; Ait Tchakoucht, T.; Saheed, Y.K. A systematic literature review for network intrusion detection system (IDS). *Int. J. Inf. Secur.* **2023**, *22*, 1125–1162. [CrossRef]
23. Breitbart, Y.; Garofalakis, M.; Martin, C.; Rastogi, R.; Seshadri, S.; Silberschatz, A. Topology discovery in heterogeneous IP networks. In Proceedings of the IEEE INFOCOM 2000, Tel Aviv, Israel, 26–30 March 2000; pp. 265–274. [CrossRef]
24. Bejerano, Y.; Breitbart, Y.; Garofalakis, M.; Rastogi, R. Physical topology discovery for large multisubnet networks. In Proceedings of the IEEE INFOCOM 2003, San Francisco, CA, USA, 30 March–3 April 2003; pp. 342–352. [CrossRef]
25. Zhou, S.; Cui, L.; Fang, C.; Chai, S. Research on network topology discovery algorithm for Internet of Things based on multi-protocol. In Proceedings of the 2018 10th International Conference on Modelling, Identification and Control (ICMIC), Guiyang, China, 2–4 July 2018; pp. 1–6. [CrossRef]
26. Pandey, S.; Choi, M.-J.; Won, Y.J.; Hong, J.W.-K. SNMP-based enterprise IP network topology discovery. *Int. J. Netw. Manag.* **2011**, *21*, 169–184. [CrossRef]
27. Tu, T.; Liu, X.; Song, L.; Zhang, Y. Understanding real-world concurrency bugs in Go. In Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '19), Providence, RI, USA, 13–17 April 2019; pp. 865–878. [CrossRef]
28. Burns, B.; Grant, B.; Oppenheimer, D.; Brewer, E.; Wilkes, J. Borg, Omega, and Kubernetes. *Commun. ACM* **2016**, *59*, 50–57. [CrossRef]
29. Gartner, Inc. Gartner Unveils Top Eight Cybersecurity Predictions for 2023–2024. Press Release, 28 March 2023. Available online: https://www.gartner.com/en/newsroom/press-releases/2023-03-28-gartner-unveils-top-8-cybersecurity-predictions-for-2023-2024 (accessed on 16 October 2026).
30. van Acken, J.-P.; Gadellaa, J.F.; Jansen, S.; Labunets, K. Poster: The unknown unknown: Cybersecurity threats of shadow IT in higher education. In Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), Copenhagen, Denmark, 26–30 November 2023; pp. 3633–3635. [CrossRef]
31. Bhuse, V.; Vellaboina, Y.; Wang, X. Enhancing network security: Rogue switch detection and prevention in local area network. *Eur. Conf. Cyber Warf. Secur.* **2024**, *23*, 66–73. [CrossRef]
32. NeDi—Network Discovery Suite. Available online: https://www.nedi.ch (accessed on 16 October 2026).
33. Netdisco—Network Management Tool. Available online: https://netdisco.org (accessed on 16 October 2026).
34. Wei, W.; Suh, K.; Wang, B.; Gu, Y.; Kurose, J.; Towsley, D. Passive online rogue access point detection using sequential hypothesis testing with TCP ACK-pairs. In Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference (IMC '07), San Diego, CA, USA, 24–26 October 2007; pp. 365–378. [CrossRef]

35. National Cyber Security Centre (NCSC). Shadow IT: Managing 'Unknown Assets' That Are Used Within an Organisation. 2023. Available online: https://www.ncsc.gov.uk/guidance/shadow-it (accessed on 16 October 2026).
36. Anderson, B.; McGrew, D. OS fingerprinting: New techniques and a study of information gain and obfuscation. *arXiv* **2017**, arXiv:1706.08003.
37. Laštovička, M.; Husák, M.; Velan, P.; Jirsík, T.; Čeleda, P. Passive operating system fingerprinting revisited: Evaluation and current challenges. *Comput. Netw.* **2023**, *229*, 109782. [CrossRef]
38. Sheng, C.; Zhou, W.; Han, Q.-L.; Ma, W.; Zhu, X.; Wen, S.; Xiang, Y. Network traffic fingerprinting for IIoT device identification: A survey. *IEEE Trans. Ind. Inform.* **2025**, *21*, 3541–3554. [CrossRef]
39. Feng, Y.; Zhang, Y.; He, H.; Zhang, W.; Wang, D. An IoT device identification method using extracted fingerprint from sequence of traffic grayscale images. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 5737–5754. [CrossRef]
40. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 4th ed.; MIT Press: Cambridge, MA, USA, 2022.
41. Gujral, H.; Sharma, A.; Jain, P.; Juneja, S.; Mittal, S. Design and implementation of a quantitative network health monitoring and recovery system. *Wirel. Pers. Commun.* **2022**, *125*, 367–397. [CrossRef] [PubMed]
42. FIRST. *Common Vulnerability Scoring System v3.1: Specification Document*; Forum of Incident Response and Security Teams: Cary, NC, USA, 2019. Available online: https://www.first.org/cvss/v3.1/specification-document (accessed on 16 October 2026).
43. Khelifati, A.; Khayati, M.; Dignös, A.; Difallah, D.; Cudré-Mauroux, P. TSM-Bench: Benchmarking time series database systems for monitoring applications. *Proc. VLDB Endow.* **2023**, *16*, 3363–3376. [CrossRef]
44. Bovenzi, G.; Cerasuolo, F.; Ciuonzo, D.; Di Monda, D.; Guarino, I.; Montieri, A.; Persico, V.; Pescapé, A. Mapping the Landscape of Generative AI in Network Monitoring and Management. *IEEE Trans. Netw. Serv. Manag.* **2025**, *22*, 2441–2472. [CrossRef]
45. Miuccio, L.; Riolo, S.; Samarakoon, S.; Bennis, M.; Panno, D. On Learning Generalized Wireless MAC Communication Protocols via a Feasible Multi-Agent Reinforcement Learning Framework. *IEEE Trans. Mach. Learn. Commun. Netw.* **2024**, *2*, 298–317. [CrossRef]
46. IEEE Registration Authority. MAC Address Block Registries: MA-L, MA-M and MA-S. Available online: https://standards.ieee.org/products-programs/regauth/mac/ (accessed on 16 October 2026).
47. Norseth, K.; Bell, E. (Eds.) *Definitions of Managed Objects for Bridges; RFC 4188*; IETF: Fremont, CA, USA, 2005. [CrossRef]
48. Levi, D.; Harrington, D. *Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering, and Virtual LAN Extensions*; RFC 4363; IETF: Fremont, CA, USA, 2006. [CrossRef]
49. Zabbix SIA. Network Discovery. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/discovery/network_discovery (accessed on 16 October 2026).
50. Paessler. Maps. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/maps.htm (accessed on 16 October 2026).
51. Paessler. Auto-Discovery. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/auto_discovery.htm (accessed on 16 October 2026).
52. SolarWinds. Network Topology Mapper Administrator Guide. NTM 2.3 Archived Guide, 57 PDF Pages. Available online: https://documentation.solarwinds.com/archive/pdf/NTM/ntm_product_guide_2-3.pdf (accessed on 16 October 2026).
53. Cisco Systems. Add and Organize Devices. Prime Infrastructure 3.10 User Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/add_and_organize_devices.html (accessed on 16 October 2026).
54. Cisco Systems. Visualize the Network Topology. Prime Infrastructure 3.10 User Guide; Chapter Updated 13 February 2024. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/visualize_the_network_topology.html (accessed on 16 October 2026).
55. Zabbix SIA. Standardized Templates for Network Devices. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/config/templates_out_of_the_box/network_devices (accessed on 16 October 2026).
56. Paessler. Monitoring via SNMP. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/monitoring_via_snmp.htm (accessed on 16 October 2026).
57. Cisco Systems. Monitor Device and Network Health and Performance. Prime Infrastructure 3.10 User Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/monitor_device_and_network_health_and_performance.html (accessed on 16 October 2026).
58. Cisco Systems. Prime Infrastructure 3.10 Release Notes. 3.10; Updated 11 April 2024. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/release/notes/cisco_prime_infrastructure_3_10_release_notes.html (accessed on 16 October 2026).
59. Cisco Systems. Configure Controller and AP Settings. Prime Infrastructure 3.10 Administrator Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/admin/cisco_prime_infrastructure_3_10_admin_guide/configure_controller_AP_settings.html (accessed on 16 October 2026).

60. Zabbix SIA. Zabbix Agent Item Keys. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/config/items/itemtypes/zabbix_agent (accessed on 16 October 2026).
61. Paessler. System Information. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/system_information.htm (accessed on 16 October 2026).
62. Cisco Systems. Manage Device Software Images. Prime Infrastructure 3.10 User Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/manage_device_software_images.html (accessed on 16 October 2026).
63. Zabbix SIA. Service Monitoring. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/it_services (accessed on 16 October 2026).
64. Paessler. Business Process Sensor. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/business_process_sensor.htm (accessed on 16 October 2026).
65. Cisco Systems. User Permissions and Device Access. Prime Infrastructure 3.10 Administrator Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/admin/cisco_prime_infrastructure_3_10_admin_guide/user_permissions_and_device_access.html (accessed on 16 October 2026).
66. Zabbix SIA. What is New in Zabbix 7.0.0. Zabbix 7.0.0 Release Documentation. Available online: https://www.zabbix.com/documentation/7.0/en/manual/introduction/whatsnew700 (accessed on 16 October 2026).
67. Paessler. Available Licenses. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/available_licenses.htm (accessed on 16 October 2026).
68. SolarWinds. NTM 2.3 Release Notes. NTM 2.3; Released 17 January 2024; Updated 5 June 2024. Available online: https://documentation.solarwinds.com/en/success_center/ntm/content/release_notes/ntm_2-3_release_notes.htm (accessed on 16 October 2026).
69. SolarWinds. NTM Product and Licensing Offer. Current Vendor Page; Licensing Corroboration Only. Available online: https://www.solarwinds.com/network-topology-mapper (accessed on 16 October 2026).
70. Cisco Systems. Prime Infrastructure 3.x Ordering and Licensing Guide. August 2020; C97-735996-04; Referenced by 3.10 Release notes. Available online: https://www.cisco.com/c/dam/en/us/products/collateral/cloud-systems-management/prime-infrastructure/presentation-c97-735996.pdf (accessed on 16 October 2026).
71. Zabbix SIA. Official Telegram Webhook README. Official Release/7.0 Branch; Requires 7.0+. Available online: https://raw.githubusercontent.com/zabbix/zabbix/release/7.0/templates/media/telegram/README.md (accessed on 16 October 2026).
72. Paessler. PRTG Telegram Configuration Guide. Official Paessler Guide. Available online: https://blog.paessler.com/how-to-receive-prtg-notifications-by-telegram (accessed on 16 October 2026).
73. Paessler. Notification Templates. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/notification_templates.htm (accessed on 16 October 2026).
74. Cisco Systems. Monitor Alarms and Events. Prime Infrastructure 3.10 User Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/monitor_alarms_and_events.html (accessed on 16 October 2026).
75. Zabbix SIA. Scheduled Reports. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/config/reports (accessed on 16 October 2026).
76. Paessler. Reports. PRTG Network Monitor Online Manual. Available online: https://manuals.paessler.com/reports.htm (accessed on 16 October 2026).
77. Zabbix SIA. SSH Agent. Zabbix 7.0 Maintained Manual. Available online: https://www.zabbix.com/documentation/7.0/en/manual/config/items/itemtypes/ssh_checks (accessed on 16 October 2026).
78. Paessler. SSH Remote Ping Sensor. PRTG Online Manual; Page Has No Build Label. Available online: https://manuals.paessler.com/ssh_remote_ping_sensor.htm (accessed on 16 October 2026).
79. Cisco Systems. Cisco Prime Infrastructure User Interface Reference. Prime Infrastructure 3.10 User Guide. Available online: https://www.cisco.com/c/en/us/td/docs/net_mgmt/prime/infrastructure/3-10/user/guide/ciscoprimeinfrastructure_3_10_userguide/user_interface_reference.html (accessed on 16 October 2026).
80. Amdahl, G.M. Validity of the single processor approach to achieving large scale computing capabilities. In Proceedings of the AFIPS Spring Joint Computer Conference, Atlantic City, NJ, USA, 18–20 April 1967; pp. 483–485. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.