Campus Network Monitoring and Incident Detection with LibreNMS: A Practical Case Study and Operational Engineering Perspective at a University Network

Barış Demirtaş
Information Technologies Directorate
TOBB University of Economics and Technology, Ankara, Türkiye
barneo@etu.edu.tr
DOI: https://doi.org/10.5281/zenodo.21115187
Article type: Case study + operational engineering paper

Abstract—University campus networks have become mission-critical infrastructures that support education, research, administrative operations, security systems, remote access, online examinations, and digital services. As the number of connected infrastructure components increases, user-reported troubleshooting alone becomes insufficient for timely fault detection and operational visibility. This paper presents a practical case study on the deployment and operational use of LibreNMS in a university network environment. The monitored environment includes more than 1,000 assets such as switches, routers, wireless access points, UPS devices, servers, firewalls, and large network printers. LibreNMS is used for SNMP-based monitoring, ICMP reachability checks, interface utilization tracking, alerting, reporting, and incident-oriented operational analysis. Operational observations show that a five-minute polling model combined with actionable e-mail alerts enables incident detection within approximately 1–5 minutes and engineering response within approximately 5–10 minutes for selected infrastructure events. The paper positions LibreNMS as an effective open-source infrastructure visibility layer when it is securely configured, properly tuned, and integrated into daily network operations. It also discusses limitations, security considerations, and practical recommendations for campus-scale monitoring deployments.

Keywords—LibreNMS, network monitoring, incident detection, SNMP, campus network, infrastructure monitoring, higher education IT, operational engineering, alerting, availability monitoring

Publication Note and Scope

This manuscript is prepared as a publishable case study and operational engineering paper. It intentionally avoids exposing internal IP addressing, complete topology diagrams, firewall policies, user information, and exact management-plane details. The purpose is to describe the engineering approach, monitoring methodology, operational impact, and lessons learned rather than to publish sensitive infrastructure data.

1. Introduction

Modern universities depend on stable, observable, and resilient network infrastructure. Academic services, learning platforms, library systems, student information systems, camera networks, wireless connectivity, VoIP, VPN, data center services, and administrative applications all rely on the campus network. A local distribution switch outage, a failing uplink, a power event, or an overloaded interface may therefore affect many independent services at the same time.

In many operational environments, the first indication of a failure is still a user complaint. This reactive model is simple, but it creates delayed detection, incomplete situational awareness, and inconsistent incident documentation. For a campus network operations team, the more sustainable model is continuous infrastructure monitoring supported by actionable alerting, historical graphing, and repeatable response procedures.

LibreNMS was selected as the central infrastructure monitoring platform in this case study because it provides automatic discovery, SNMP-based monitoring, customizable alerting, graphical historical data, and broad device support. The official LibreNMS project describes the platform as a full-featured network monitoring system with automatic discovery mechanisms such as CDP, LLDP, OSPF, BGP, SNMP, and ARP, together with flexible alerting and API capabilities [1].

The contribution of this paper is not a laboratory benchmark. Instead, it is an operational engineering case study based on live campus monitoring practice. The focus is on how LibreNMS was positioned inside daily operations to support faster incident detection, reporting, visibility, and infrastructure renewal planning.

2. Operational Context and Motivation

The monitored environment is a university-scale infrastructure with more than 1,000 monitored assets. The environment includes active network devices, server infrastructure, power-related devices, firewall components, access infrastructure, and large network printers. The operational challenge is not simply knowing whether one device is reachable; it is understanding the blast radius of failures, the history of recurring events, and the relationship between infrastructure conditions and service impact.

Operational requirement	Monitoring objective
Early outage detection	Detect device-down, interface-down, and power-related events before or at the earliest stage of user impact.
Infrastructure visibility	Provide a single operational view for network, server, UPS, firewall, AP, and printer infrastructure.
Root-cause analysis	Correlate symptoms such as camera outage, wireless outage, or network socket outage with upstream infrastructure events.
Capacity planning	Use historical utilization graphs to support link upgrade, device renewal, and topology improvement decisions.
Daily reporting	Produce operational reports that support management visibility and system renewal planning.
Operational discipline	Move from user-report-based troubleshooting to monitored, timestamped, and documented incident handling.

Table 1. Operational monitoring requirements and objectives

3. LibreNMS Deployment Environment

The LibreNMS deployment is operated as a centralized monitoring node. The operational snapshot below was obtained through a live read-only check. Sensitive values such as internal IP addresses and complete host-management details are redacted for the public version of this paper.

Field	Observed value
Host	LibreNMS monitoring node; exact hostname redacted for publication
Management IP	Private RFC1918 management address; redacted for publication
Operating system	Debian GNU/Linux 11 (bullseye)
Kernel	5.10.0-33-amd64
CPU	CPU model not exposed by allowlist during read-only check; LibreNMS inventory reports Generic x86 64-bit
RAM	32145 MB total, approximately 31.4 GiB

Field	Observed value
RAM utilization snapshot	9815 MB used, 21501 MB available
Swap	4095 MB total, 737 MB used
Root disk	/dev/sda1, 246 GB
Disk utilization snapshot	118 GB used, 118 GB available, 51% utilization
Uptime at check time	120 days, 11:34
Load average	0.27, 1.40, 1.75
Average polling interval	Approximately 5 minutes

Table 2. LibreNMS server snapshot used for the case study

Design interpretation: The server resources are sufficient for the described monitoring scope when poller timing, database health, RRD storage, alert noise, and discovery processes are controlled. As the monitored estate grows further, distributed polling and storage optimization should be evaluated as planned scaling mechanisms rather than emergency measures.

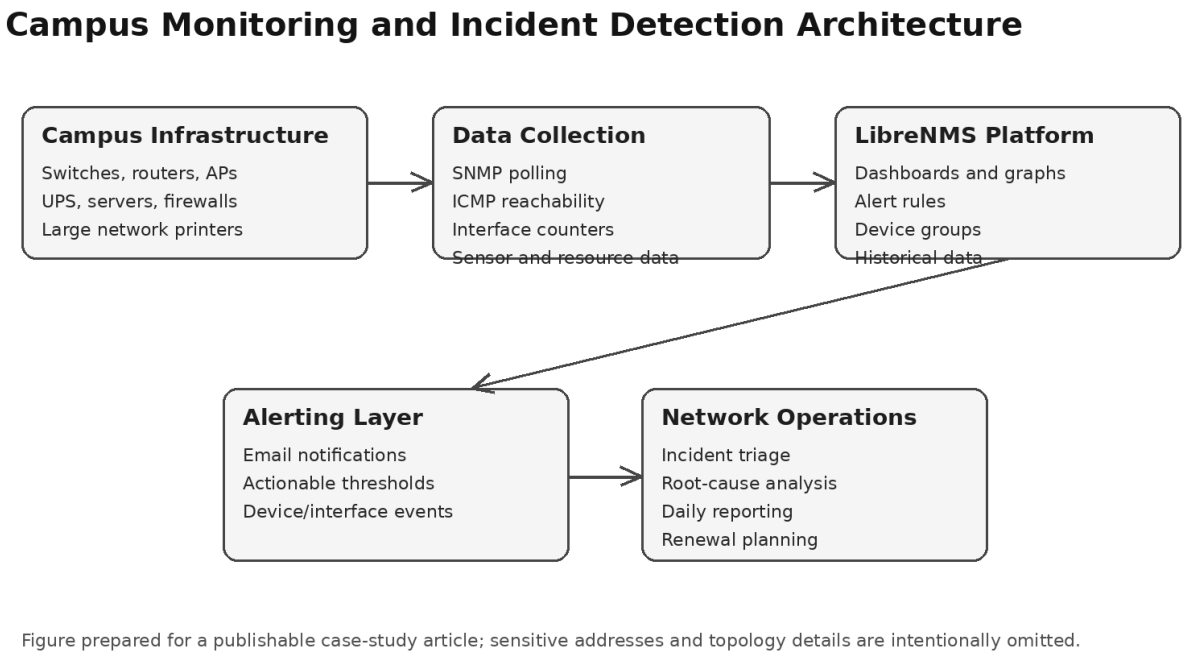

Figure 1. High-level monitoring and incident-detection architecture

4. Monitoring Scope and Data Collection Model

The monitoring scope covers more than 1,000 infrastructure assets. The system is not limited to classic network switches; it also includes infrastructure elements whose failure can affect operational continuity, security operations, or service delivery. The main monitoring methods are SNMP polling for detailed metrics and ICMP reachability checks for availability validation. LibreNMS supports adding devices through SNMP and can also monitor ping-only devices where full SNMP polling is not practical [2].

Asset category	Typical monitored data	Operational use
Switches	Reachability, interface status, interface utilization, errors, CPU, memory, sensors where available	Detect uplink loss, access-layer outages, port saturation, and distribution-level impact.
Routers	Availability, interface counters, routing-related status where available, CPU and memory	Support internet-edge and campus routing visibility.

Asset category	Typical monitored data	Operational use
Wireless access points	Reachability, controller or device status, uptime, resource status where available	Detect AP outages and local wireless-service degradation.
UPS devices	Reachability, battery or power status where available, environmental data where supported	Differentiate power-related incidents from pure network failures.
Servers	Availability, resource status through SNMP or agent-supported methods where applicable	Monitor infrastructure service health and identify resource exhaustion.
Firewalls	Availability, interface counters, CPU, memory, session-related indicators where supported	Support perimeter availability and throughput visibility.
Large network printers	Reachability and basic device status where supported	Reduce support ambiguity for shared, high-use printing infrastructure.

Table 3. Monitoring scope and operational purpose

5. Implementation Methodology

5.1 Inventory and Criticality Grouping

Devices were first grouped according to operational criticality. Core and distribution devices were prioritized because a failure at these layers has a larger service impact than a single endpoint problem. Access-layer switches, wireless infrastructure, UPS devices, firewalls, servers, and large shared printers were then added progressively.

5.2 Secure SNMP Configuration

SNMP was used as the primary telemetry mechanism. The monitoring design assumes that polling should be permitted only from the monitoring server or monitoring subnet. SNMP communities or SNMPv3 credentials must be protected as sensitive operational secrets. Vendor-specific SNMP configuration varies; LibreNMS provides SNMP configuration examples for several platforms [3].

5.3 Discovery and Device Classification

Discovery was used to identify device type, interfaces, sensors, operating-system metadata, resource values, and supported modules. Automatic discovery is useful in large environments, but it should be combined with naming standards and device groups to avoid uncontrolled inventory growth. LibreNMS auto-discovery can use multiple network data sources and requires at least one seed device before auto-discovery works [4].

5.4 Polling and Historical Data

A practical five-minute polling interval was used as the baseline. This provides a reasonable balance between detection speed and system load. Historical interface graphs, device uptime, resource data, and event timelines provide engineering evidence during troubleshooting and capacity planning.

5.5 Alert Rule Engineering

Alert rules were designed around actionable events rather than every possible state change. LibreNMS alert rules are based on entities, conditions, and values, which makes it possible to construct rules for device availability, interface operational status, resource usage, and sensor thresholds [5].

5.6 Reporting and Operational Review

The monitoring platform supports daily reporting, recurring incident review, and system renewal discussions. The value of monitoring increases when graphs and alerts are used in operational meetings, procurement planning, and post-incident analysis rather than being left as passive dashboards.

6. Incident Detection and Response Workflow

The incident workflow was structured around early detection, verification, classification, and action. The platform alerts the responsible team through e-mail. Engineers then validate the event through device status, related device groups, interface graphs, power indicators, and known topology relationships.

Step	Activity	LibreNMS evidence	Operational outcome
1	Abnormal condition occurs	Device down, interface down, utilization threshold, resource threshold	Incident candidate is created.
2	Alert is triggered	E-mail notification and event timestamp	Responsible engineers are informed.
3	Engineer validates impact	Device page, related ports, graphs, neighboring devices, device group	False positives and duplicate symptoms are filtered.
4	Root cause is classified	Availability timeline, port counters, power-related device status where available	Incident is categorized as power, uplink, device, configuration, saturation, or service-related.
5	Action is taken	Current status and historical graph comparison	Technical team performs corrective action or escalates to facility/security/vendor teams.
6	Incident is documented	Alert timestamp, response time, graph evidence	Operational knowledge base and daily reports are improved.

Table 4. Incident detection and response workflow

Observed timing: For selected infrastructure events, incidents were noticed within approximately 1–5 minutes, and technical action was initiated within approximately 5–10 minutes. These values should be interpreted as operational observations rather than controlled laboratory measurements. Detection time is naturally affected by polling interval, alert rule timing, e-mail delivery, and engineer availability.

7. Practical Incident Observations

7.1 Building Distribution Switch Power Interruption

One critical incident involved an energy interruption affecting a building distribution switch. The outage caused the distribution switch to become unreachable and produced downstream visibility loss for multiple dependent devices. From an operational perspective, this distinction was important: the event was not interpreted as many independent endpoint failures, but as a single distribution-level incident with a broader service impact. The monitoring view helped identify the affected zone, the approximate start time, and the dependent infrastructure that appeared offline as a consequence.

7.2 Wireless Access Point Interruption

Wireless access point interruptions were monitored through reachability and infrastructure status indicators. Wireless issues are often reported by users as “Wi-Fi is not working,” but monitoring helps distinguish between a local access point issue, upstream switch issue, controller-side issue, or wider distribution-layer incident. This reduces time spent on blind troubleshooting.

7.3 Port Utilization Thresholds

Interface utilization thresholds were used to detect abnormal or sustained port usage on switches. This is useful for identifying saturated uplinks, unexpected traffic patterns, misbehaving endpoints, or capacity bottlenecks. Historical graphs are particularly valuable because they show whether the condition is sudden, recurring, or gradually increasing over time.

8. Results and Operational Impact

The deployment changed the operational model from primarily user-reported troubleshooting to monitoring-driven infrastructure visibility. The most important improvement was not only earlier alarms, but also the availability of timestamped evidence, historical data, and recurring reports.

Dimension	Before LibreNMS-centered monitoring	After LibreNMS operational integration
Detection model	User notification was often the first signal.	E-mail alerts and dashboard status provide earlier technical visibility.
Incident evidence	Troubleshooting depended heavily on manual checks and device login.	Graphs, timestamps, device status, interface counters, and historical values support root-cause analysis.
Response time	Response started after complaint or manual discovery.	Selected incidents were noticed within 1–5 minutes and action was initiated within 5–10 minutes.
Reporting	Incident reporting was more manual and fragmented.	Daily reports and graph-based summaries support operational review.
Capacity planning	Upgrade discussions relied more on observation and user impact.	Interface history and device trends support renewal and capacity decisions.
Management visibility	Technical status was harder to summarize quickly.	Monitoring dashboards and reports provide a clearer infrastructure picture.

Table 5. Operational impact summary

9. Discussion

LibreNMS is most effective when treated as an operational platform rather than a software installation. The difference is important. Installing a monitoring system creates dashboards; engineering a monitoring practice creates visibility, actionability, and accountability.

For campus networks, the strongest value appears in three areas. First, device-down and interface-down alerts help identify outages earlier. Second, interface utilization and resource graphs support technical diagnosis and capacity planning. Third, daily reporting converts monitoring data into management communication and system renewal evidence.

At the same time, monitoring introduces its own engineering responsibilities. Alert rules must be tuned. Device naming must be consistent. Access to monitoring data must be protected. SNMP credentials must be handled

carefully. Poller health, disk usage, RRD storage, database performance, and application updates must be reviewed regularly. LibreNMS performance documentation recommends RRDCached to reduce I/O load, and scaling mechanisms such as distributed polling should be considered when device count, latency, or polling workload grows [6], [7].

10. Security and Privacy Considerations

- Restrict SNMP access to the LibreNMS server or approved monitoring subnet.
- Prefer SNMPv3 where vendor support and operational maturity allow it.
- Do not expose the LibreNMS web interface directly to the public internet.
- Use strong authentication, role-based access, and administrative audit discipline.
- Avoid publishing internal IP addresses, full topology diagrams, firewall policies, VLAN information, or device names in public articles.
- Protect e-mail alerting credentials and any webhook or API tokens.
- Back up the LibreNMS database, configuration, and critical operational customizations.
- Keep the operating system, web stack, database, and LibreNMS application updated through a controlled change process.

11. Limitations

- This paper is based on operational case-study observations rather than a controlled experimental design.
- The incident detection and response times are approximate operational ranges, not statistically validated averages.
- Exact topology, device names, internal IP addresses, and sensitive configuration details are intentionally omitted.
- The monitored environment, vendor mix, firmware versions, SNMP implementation quality, and organizational process maturity may affect reproducibility.
- LibreNMS should not be positioned as a complete replacement for SIEM, NetFlow/IPFIX analytics, log management, vulnerability management, or endpoint security platforms. It is best understood as an infrastructure visibility and availability monitoring layer.

12. Recommendations for Campus Deployments

Area	Recommendation
Deployment order	Start with core, distribution, firewall, UPS, and server infrastructure before expanding aggressively into access-layer details.
Device naming	Use consistent hostnames, locations, and interface descriptions; monitoring quality depends heavily on naming discipline.
Alert engineering	Define actionable alerts. Do not alert on every access port flap or every non-critical event.
Dependency thinking	Interpret access-device outages in relation to upstream distribution switches, power systems, fiber paths, and firewall dependencies.
Reporting	Use daily reports to summarize incidents, device health, capacity risks, and renewal priorities.

Area	Recommendation
Security	Redact sensitive information in public reports; restrict access to the monitoring platform and SNMP credentials.
Scaling	Monitor poller completion, database health, disk growth, and RRD storage; evaluate RRDCached and distributed polling before performance becomes a crisis.
Continuous improvement	Review false positives and missed events after each incident; update alert rules and operational playbooks accordingly.

Table 6. Recommendations for university-scale LibreNMS deployments

13. Conclusion

This case study shows that LibreNMS can provide strong operational value in a university campus network when it is deployed as part of a disciplined monitoring and incident-response process. In the described environment, more than 1,000 assets are monitored across network, server, power, firewall, wireless, and shared-device categories. A five-minute polling approach combined with e-mail alerting enables selected incidents to be noticed within approximately 1–5 minutes and acted upon within approximately 5–10 minutes.

The most important result is the shift from user-reported troubleshooting to monitoring-driven operational visibility. LibreNMS provides early detection, graph-based evidence, daily reporting, and infrastructure renewal support. However, these benefits require careful SNMP security, alert tuning, performance monitoring, data protection, and operational ownership.

For higher education institutions, open-source monitoring can be a practical and cost-effective foundation for network operations. The success factor is not merely installing LibreNMS; it is integrating the platform into daily engineering practice, incident communication, capacity planning, and management reporting.

References

[1] LibreNMS, “LibreNMS — Features and Overview,” official project website. https://www.librenms.org/

[2] LibreNMS Documentation, “Adding a Device.” https://docs.librenms.org/Support/Adding-a-Device/

[3] LibreNMS Documentation, “SNMP Configuration Examples.” https://docs.librenms.org/Support/SNMP-Configuration-Examples/

[4] LibreNMS Documentation, “Auto-discovery Setup.” https://docs.librenms.org/Extensions/Auto-Discovery/

[5] LibreNMS Documentation, “Alerting Rules.” https://docs.librenms.org/Alerting/Rules/

[6] LibreNMS Documentation, “Performance Optimisations.” https://docs.librenms.org/Support/Performance/

[7] LibreNMS Documentation, “Distributed Polling.” https://docs.librenms.org/Extensions/Distributed-Poller/

[8] LibreNMS Documentation, “Setting up RRDCached.” https://docs.librenms.org/Extensions/RRDCached/

[9] LibreNMS Documentation, “Network Map.” https://docs.librenms.org/Extensions/Network-Map/

[10] LibreNMS Documentation, “Authentication Options.” https://docs.librenms.org/Extensions/Authentication/

Author Note

Prepared by: Barış Demirtaş, Information Technologies Directorate, TOBB University of Economics and Technology. Before public submission, institutional approval and anonymization level should be reviewed.