Uplivra is in beta and online purchasing is paused. **Apply to test it**: testers get a license on request and 10% off when purchasing opens.

INSTALL GUIDE · NETWORK ENGINEERS, IT TEAMS AND MSPS

Network Pro in depth

Restore saved configurations safely, plan capacity from real traffic, see VLANs, PoE and optic health on every port, follow what moved on the network, test the links between your sites, and know the limits.

Uplivra Technologies LLC · Guide for Uplivra 26.10 · Updated 28 September 2026 · Latest version: https://uplivra.com/guides/network-pro.html

What you need

- Uplivra 26.10.30 or later, with **Network Pro**. Everything on this page needs Network Pro; nothing here is in Core.
- Switches and routers monitored with an **SNMP check** (for ports, VLANs, PoE, optics, capacity and topology).
- For restores: a **Configuration backup** check on the device and the **Restore saved configurations to devices** permission for two people.

Restore a saved configuration

Configuration backups keep a version each time a device's configuration changes. You can put an older version back on the device.

Which devices. Only device families that can replace the whole configuration at once **and undo it by themselves** if something goes wrong:

Device	How Uplivra restores	What the device needs
Cisco IOS / IOS-XE	Copies the file to `flash:` with SCP, then `configure replace flash:file force revert trigger error timer N`, and `configure confirm` over a new connection	`ip scp server enable`, and an `archive` path (the undo timer needs it)
Arista EOS	Copies the file to flash with SCP, replaces a configuration session from it (`rollback clean-config`, `copy flash:file session-config`), `commit timer`, then `commit` over a new connection	SCP allowed for the login
Juniper Junos	Copies the saved `set` commands to `/var/tmp/` with SFTP, then `configure exclusive`, `delete`, `load set`, `commit check`, `commit confirmed N`, and `commit` over a new connection	SFTP allowed for the login

Every other device type is refused, with the reason on the page. Files are copied only with SCP or SFTP, checking the device's SSH host key. Never TFTP, FTP or Telnet.

Steps.

1. Open **Network › Configuration backups**, pick the device, click a version and press **Restore this version**.
2. **Check the device now.** Uplivra reads the device's configuration through its collector. The restore stops if the device doesn't answer, answers with a different SSH host key, or answers as a different device type. If the device **no longer matches its newest backup** (someone changed it since), you see exactly what differs and must tick a box to go on: the restore replaces those changes too. You also see every line the restore changes.
3. **Send for approval.** Say why, and choose whether to **wait for a maintenance window** covering the device.
4. **Another person approves.** Someone else with the restore permission approves or turns it down under **Network › Configuration backups › Restores**. The person who asked can't approve their own. While it waits, an alert stays open.

5. **The restore runs** right after the approval, or when a maintenance window covering the device begins. The collector checks once more that the device still has the configuration that was approved (if not, it stops without changing anything), copies the file, applies it with an undo timer (2-60 minutes, 10 by default), reconnects and confirms. If it can't reconnect, the device undoes the change by itself when the timer runs out.
6. **Read back.** A fresh backup is saved and compared with the version you restored. Devices sometimes add default lines; they show as differences.

Requests not sent within a day, or not approved (or not started by a window) within 7 days, expire. Every step, including who looked at the differences, is in the audit log under `config.restore`.

Permission. *Restore saved configurations to devices, and approve others' restores* (`configs.restore`) is a sensitive permission. Administrators have it. It includes seeing configuration backups.

Capacity reports

Network › Capacity lists every interface your SNMP checks measure:

- **95th percentile, in and out, over 7, 30 and 90 days.** Uplivra takes the 5-minute averages of the traffic and reports the value that 95% of them stay at or below: a fair "how busy is it really" that ignores short bursts. Next to it, the share of the link speed.
- **Growth.** Each day's busy hour (the highest hourly average of the busier direction) over up to 90 days, with a straight line fitted through them by least squares. Shown per 30 days, in bits per second and as a percentage. It needs at least 7 days of data.
- **Days to 80% and 100%.** When the line reaches 80% and 100% of the link speed at that pace. "—" means the traffic isn't growing or the link speed isn't known; "already" means it's there now. Interfaces reaching 80% within 90 days are highlighted.

Filter by site, device or name, sort by any column (soonest to fill first by default) and **Download CSV** for a spreadsheet with the same figures in bits per second.

The report uses the 5-minute and hourly averages Uplivra keeps for 90 days and 2 years (see **Settings › Retention**); shorten those and the longer windows have less to go on.

VLANs, PoE and optics on every port

A switch's device page lists its ports. With Network Pro they also show:

Tagged VLANs. The **VLAN** column shows the untagged (native) VLAN and, below it, the VLANs the port carries tagged, written compactly (`10, 20-29, 99`). Uplivra reads them from the standard Q-BRIDGE-MIB (which ports carry each VLAN, and which untagged; the current table, or the configured one when a switch has no current table). Cisco switches don't fill Q-BRIDGE-MIB, so on Cisco Uplivra reads the trunk VLAN lists and native VLAN (CISCO-VTP-MIB) and each access port's VLAN (CISCO-VLAN-MEMBERSHIP-MIB).

PoE watts per port. The **PoE** column shows the watts each port delivers, and the page adds them up (**PoE power delivered**), next to the switch's budget. Per-port watts come from:

Vendor	Where Uplivra reads it
Cisco (Catalyst, IE, and others with PoE)	CISCO-POWER-ETHERNET-EXT-MIB `cpeExtPsePortPwrConsumption`
MikroTik	MIKROTIK-MIB `mtxrPOEPower`
Others (HPE Aruba, Juniper, Ubiquiti, Netgear and so on)	On/off, faults and the switch's total from the standard POWER-ETHERNET-MIB. Their per-port watts aren't in a MIB Uplivra reads yet

Each port's watts are also kept as the **PoE power on the port** measurement, so you can chart it and set alert rules on it.

Optic health. For ports with an optical module (SFP, SFP+, QSFP) that reports digital optical monitoring, the **Optic** column shows receive and transmit light (dBm), module temperature, supply voltage and laser bias current. Uplivra reads the sensors from ENTITY-SENSOR-MIB or CISCO-ENTITY-SENSOR-MIB and ties each to its port through ENTITY-MIB (or the sensor's name, such as "Te1/1/1 Module Temperature Sensor" or "DOM Rx Power Sensor for Ethernet49/1"). Arista, Cisco and many others report these.

Alerts on optics. Where the device publishes its own limits (Cisco's sensor thresholds), a crossed limit shows on the port ("high temperature (limit 70 °C), major") and counts as a problem. Two alert rules are added for you: **Optic out of range (device's own limits)** and **Optic running hot** (above 70 °C for 15 minutes). The readings are kept as measurements (**Optic temperature**, **Optic supply voltage**, **Optic laser bias**, **Optic out of range**) so you can chart them and add your own rules, alongside the existing **Receive light low** rule.

Topology change history

Every time a switch or router reports its LLDP and CDP neighbors (every few minutes), Uplivra compares it with the last report and keeps what changed. **Network › Network map › What changed** lists it, newest first:

- **New:** a neighbor appeared on a port (with the neighbor's own port).
- **Gone:** a neighbor disappeared from a port.
- **Moved:** the same neighbor now on another port of the same switch, or gone from one switch and turned up on another within a day ("ap-lobby moved from sw-a Gi1/0/9 to sw-b Gi0/3").
- **MAC moved:** a device, by its MAC address, moved to another switch port. Only access ports with a few devices on them count, not uplinks, so the list isn't flooded when traffic takes another path.

Filter by device, by neighbors or MAC moves, and by the last day, 7 days, 30 days or year. A switch's device page links to its own history (**What moved on this device's ports**), which also lists what moved away from it. The first report from a device logs nothing, and a year of history is kept.

Site-to-site tests

A site-to-site test measures the network **between two of your sites**: a collector at one site sends small UDP probes to a collector at the other, which sends each one straight back.

1. Open **Network › Site-to-site tests** and fill in **Add a test**: the sending site, the collector that answers, and that collector's address **as the sending site reaches it** (its VPN or WAN address; the addresses it reports are listed next to it).
2. Pick the **UDP port** (4717 unless you change it), how many probes each run sends (50), how far apart (20 ms, like a voice call), when loss should warn (2%), and how often to run (every 5 minutes).
3. Allow that UDP port from the sending collector to the answering one in any firewall or VPN policy between the sites.

The answering collector opens the port only while a test is aimed at it, closes it when the last one is removed, and answers only probes signed with that test's key (an HMAC with a random key made for each test), never with more than it received, and at most 2,000 answers a second. Probes are 172 bytes, the size of a G.711 voice packet.

What you get. Each run gives:

- **Loss:** probes that didn't come back, as a percentage.

- **Latency:** the average round trip, with the lowest and highest.
- **Jitter:** the average change in round-trip time between one probe and the next.
- **Voice quality (MOS):** an estimate on the usual 1-5 scale (4.0 and up is good, 3.6-4.0 fair, below 3.6 poor).

Each test is a check on a stand-in device named "Site test to ..." at the sending site (it isn't counted as a device on your license), so results are kept, charted and alerted on like other checks, and show on **Path quality** too. The test goes down when nothing answers, and warns at the loss you chose; the **Voice quality poor** rule warns when MOS stays below 3.6.

How MOS is worked out. Uplivra uses the simplified ITU-T G.107 E-model common in network monitoring tools, for G.711:

1. Effective delay (ms) = average round trip ÷ 2 + 2 × jitter + 10.
2. R = 93.2 − effective delay ÷ 40 when the effective delay is under 160 ms; otherwise R = 93.2 − (effective delay − 120) ÷ 10.
3. R = R − 2.5 × loss (%), kept between 0 and 100.
4. MOS = 1 + 0.035 × R + 0.000007 × R × (R − 60) × (100 − R), at least 1.

It's an estimate from the network's behaviour, not a measurement of real calls. Ping checks use the same method.

Limits

Network Pro publishes and enforces these limits. What goes over them is dropped **and counted**, never silently lost. **Network › Traffic flows › Limits** shows the limits and, for the last 24 hours, what each collector received and dropped.

What	Limit	Over the limit
Flow records (NetFlow, IPFIX and sFlow together)	**20,000 records a second per collector**, with up to a second's worth taken at once	Dropped at the collector and counted per minute. An alert opens while a collector drops records ("Flow records dropped on collector ...")

What	Limit	Over the limit
Conversations kept	The top 300 conversations per exporter per minute (up to 100,000 distinct conversations a minute are added up)	The rest are added up as "other", so totals stay right
Server out of reach	A collector keeps 60 minutes of flow summaries	Older minutes are dropped and counted
Event viewer (logs from network devices)	**1,000,000 messages a day** per customer (days in UTC), kept 7 days. SNMP traps don't count	Dropped and counted; an alert opens when today's limit is reached. It starts again at midnight UTC

How the flow limit was set: decoding and adding up NetFlow records takes one processor core about half a microsecond per record (measured on a 2-core machine with the `BenchmarkDecodeAggregate` benchmark: about 2.5 million records a second). The limit leaves most of the collector for its checks, logs and uploads. To stay under it, send sampled flows (1 in 100 or 1 in 1000 is typical for busy links), point fewer exporters at one collector, or add a collector.

With **Log Intelligence**, its own monthly allowance (in GB) applies to logs instead of the event viewer's daily limit. Collectors older than 26.10.30 don't report their flow counts; the page says so.