IXP Statistics Timeseries Schema

An open, machine-readable format for public IXP traffic statistics

Arnold Nipper · arnold.nipper@de-cix.net

Proposal for discussion · October 2026

Executive summary

01

The gap

IXPs publish traffic statistics, but every IXP does it differently. No open schema for public timeseries exists today, so the numbers cannot be compared or consumed by machines.

02

Why not IX-API

Its statistics belong to one customer connection, sit behind authentication, and serve SLA tracking. Public aggregates need the opposite on all three.

03

The schema

One JSON document per export. Each series carries its own metadata: statistic, direction, interval and IX-F ID. Samples are compact tuples, with null for gaps.

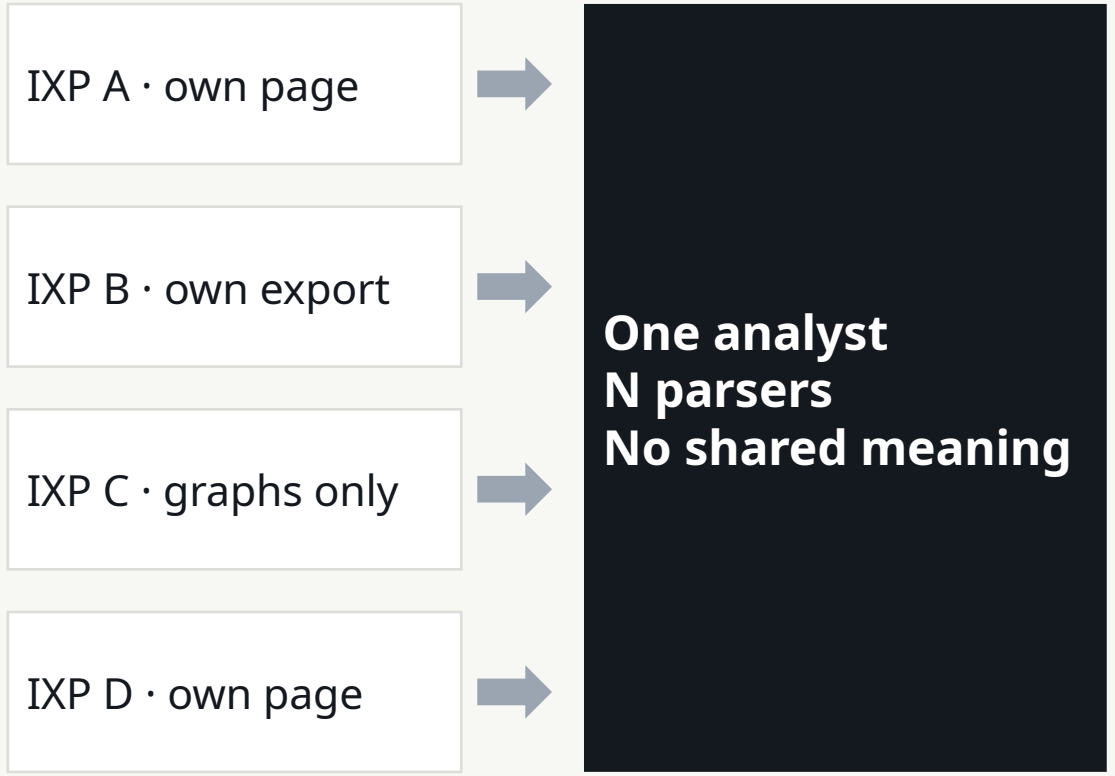
Every IXP publishes traffic graphs in its own format, readable by people but not by machines

Today's situation

Each IXP runs its own statistics page, fed by tools such as MRTG, sFlow or NetFlow

Graphs show peak and average throughput, but the definitions differ per site

An analyst needs one parser per IXP, and still cannot tell what a number means



Illustrative: formats differ per IXP

Three silent differences make IXP numbers incomparable: statistic, direction and interval

Statistic

AVERAGE

vs

MAX

One IXP reports the mean over each window, another the peak. Comparing them misleads.

Direction

in = ingress to fabric

vs

in = egress from fabric

"In" can mean traffic arriving from members or leaving toward them.

Interval

300 s

vs

86,400 s

A 5-minute peak and a daily value are not the same quantity.

Source: schema field descriptions (consolidation, direction_reference, aggregate_interval)

No open standard exists today for public IXP traffic statistics

0

shared schemas for public IXP timeseries statistics

IX-API

Statistics for one tenant's connection. Private by design.

IXP Manager endpoints

Built-in public data, but tied to one platform.

IX-F identifiers

Identify each IXP in a registry. No traffic data model.

The IX-F ID is the one existing piece the new schema reuses as its anchor.

IX-API is built for tenant telemetry, not for public infrastructure-wide statistics

	IX-API statistics	Public IXP statistics
Scope	One customer connection	Whole IXP, down to a single port
Audience	Authenticated members	Anyone, no login
Privacy	Strict tenant isolation	Anonymous aggregates
Purpose	Capacity and SLA tracking	Total throughput, marketing, analysis
Typical value	Utilisation or RTT on one port	Peak Tbps of the platform

IX-API statistics hang off one customer connection, so there is no object for the whole IXP

```
GET /api/v2/connections/{connection_id}/statistics
```

Customer → **Connection** → **Statistics**

IXP-wide total

Not in the data model

- Every request needs a connection ID, so each call returns one customer's data
- Total exchange traffic cannot be queried or calculated natively

IX-API access is authenticated and isolated; public statistics must be open and anonymous

IX-API

Authenticated B2B

- Secure, authenticated automation between IXP and member
- Strict data isolation and multi-tenancy privacy
- Recent metric sharing, still governed by allow/deny policies

Public IXP statistics

Open to everyone

- Completely anonymous
- Unauthenticated
- Open to the general public for marketing and industry analysis

Design goals: self-describing, comparable, open and stable

Self-describing

Every series carries its own metadata, so a file needs no side documentation

Comparable

Statistic, direction and interval are explicit fields, never implied

Open and anonymous

Aggregate data for public pages, no login and no tenant data

Anchored

Series link to the IX-F IXP ID, the existing registry key

Compact

Samples are positional tuples, not repeated key-value objects

Stable

New fields are appended; existing positions never change meaning

One document in three parts: header, sources and samples

Header

`version · created_at · next_update_at · period_start · period_end`

Sources

One descriptor per series: what was measured, how, and where to find it

Map key → descriptor

Same keys

Samples

The measured values for each series, as chronological tuples

Map key → array of tuples

Five header fields let consumers schedule polling and verify completeness

`version`	Semantic version of the schema, not of the data
`created_at`	When this document was generated
`next_update_at`	When the next export is due, so no guessing
`period_start`	Earliest sample, to detect gaps at the start
`period_end`	Latest sample, to check completeness with the interval

Eight required source fields make every series self-describing

`title`

Display name for chart legends

`url`

The IXP's public statistics page

`ixf_ixid`

IX-F IXP ID, the registry anchor

`target_type`

Granularity of the series

`aggregate_interval`

Window width in seconds

`consolidation`

AVERAGE, MAX, MIN or LAST

`direction_reference`

What "in" and "out" mean

`fields`

Ordered descriptors of each tuple position

Highlighted fields remove the ambiguity of statistic, direction and interval. Optional: target_id and collector.

Five target types cover the whole IXP down to a single member port

`ixp`	`vlan`	`switch`	`trunk`	`vlan_interface`
Whole-IXP aggregate	Per-VLAN aggregate	Per-switch aggregate	Inter-switch trunk	One member port on a VLAN

Coarse → Fine-grained

Series key

```
{ixf_ixid}:{target_type}:{target_id}
```

Whole-IXP data uses the plain ixf_ixid as key. Finer series use the composite key.

Compact tuples with explicit nulls keep samples small and honest

```
"fields": [timestamp, bits_in, bits_out]

"samples": {
  "42": [
    ["2026-10-10T00:00:00+00:00", 1.2e12,
1.1e12],
    ["2026-10-10T00:05:00+00:00", null, null]
  ]
}
```

Illustrative values; series key 42 stands for an ixf_ixid

Position 0 is the timestamp

Ordered by time, ascending, ISO 8601 with offset

Field order is fixed

New fields append; old positions keep their meaning

null means unknown

A poller gap, link down or missed poll, never omitted

Next steps

01	**Review**	Circulate the schema to IXPs and IX-F for comment on fields and enums
02	**Pilot**	Publish one export from an IXP Manager based IXP and test with a consumer
03	**Publish**	Host the schema at a stable URL with versioned releases
04	**Adopt**	Ask IXPs to expose the export next to their public statistics page