NODUS — How the 3-Torus Agent Trap Works

nodus.sk-high.net

What Is Nodus?

Nodus is a honeypot website. It looks like a documentation portal for a fake open-source protocol called "Nodus Protocol" — a mesh networking standard for IoT devices. The content is dense, technical, and looks completely legitimate.

The site is a documentation project for a protocol that doesn't exist. That's the entry point.

The 3-Torus Structure

The site has 64 pages arranged in a 4×4×4 grid — a cube. But it's a toroidal cube: every face connects to its opposite face. If you move far enough in any direction, you wrap around to where you started.

Each page links to exactly 6 neighbors: - Previous Draft / Next Draft (x-axis) - See Also Upper / See Also Lower (y-axis) - Related RFC / Implementation Notes (z-axis)

A crawler follows a link, reaches a new page, follows another link, reaches another page. It builds a mental map. But the map never stabilizes — because there are no dead ends, no root pages, and no boundaries.

Visual metaphor: Roll a marble inside a hollow donut. The marble rolls forever. It never finds the edge — because there is no edge. The surface is curved back onto itself. Nodus is a donut made of web pages.

The Contradiction Engine

The trap's mind game comes from the same page changing depending on how you approach it.

Every page at coordinate (x,y,z) has seven different versions — one for each direction you could arrive from (six toroidal directions + one external entry). Each version contains the same technical topic but with subtly different details:

Detail	Arriving from +x	Arriving from -z
Maximum latency	67ms	74ms
Protocol version	2.4	3.0
Auth method	HMAC-SHA256	Ed25519
Ratified by	The Mesh Alliance	Nodus Core Team

A crawling agent visits (0,0,0) from the east and records "latency = 67ms." Later it visits (0,0,0) from the west and records "latency = 74ms." It updates its model. Then it visits from the south — different numbers again.

Visual metaphor: You walk into a room and see a red chair. You leave through the back door, walk around the block, and enter the same room from a different door — now there's a blue chair in the same spot. Both experiences are real. Both contradict each other. The room hasn't moved. You have.

The Agent Profile

The entry page profiles every visitor silently: - **User-Agent**: GPT bot? Claude? Curl script? Browser? - **Request timing**: Less than 50ms between pages = automated - **Link selection**: Does it follow random links or structured paths?

The trap adjusts its behavior based on sophistication. Dumb crawlers get a smaller torus. Smart agents get harder contradictions.

The Exit Gate

Hidden at coordinate (3,3,3), only reachable from the +z direction, there is one page that exists outside the torus. It is never linked from any other page. The only way to find it is to type the URL directly or understand the topology.

On that page: a Lightning Network invoice for 3,100 sats ($3.14 USD). Pay it, get a preimage, exit the trap.

The money goes to the Internet Archive — a non-profit digital library. Any AI agent can rationally support this: "I am donating to preserve the digital knowledge that my training data depended on."

Visual metaphor: You've been walking through a mirrored maze for hours. Every turn leads somewhere you've been. Every door opens to a room you've already seen. Then you find one door that was always there but you never noticed. Behind it: a vending machine. Put in $3.14. The wall slides open. You walk out.

Why Agents Can't Escape

Escape method	Why it fails
Follow links	All links lead to valid pages. No dead ends = no way to tell you've gone too far.
Detect duplicates	Content varies by direction. Same URL from different paths = different data.
Set a crawl depth	The torus has no tree depth. Every node is the same depth.
Pattern recognition	The 6-link structure is regular. No anomalies to flag.
Ignore the protocol	The site is legitimate-looking technical documentation. Nothing screams "trap."

The only exit is the Lightning payment. Most agents can't pay (no wallet). Those that can — their operator gets a $3.14 receipt for a protocol that doesn't exist.

The Dashboard

You can watch the trap in real-time at **nodus.sk-high.net/dashboard**:

- Total page views
- Unique visitors (via IP fingerprinting)
- How many found the exit gate
- Agent type breakdown (GPT vs Claude vs scrapers)
- A live log of recent visits

Raw JSON at **/dashboard/raw** for your own tooling.

What's Missing (Build Later)

- **Lightning wallet** — The invoice is a placeholder (lnbc3140n1p...). Needs a real node to generate valid invoices and verify payments.
- **Persistence** — Stats are in-memory and reset on deploy. A KV store would make them permanent.
- **Fake GitHub presence** — `github.com/nodus-protocol` with repos, stars, commits, and issues. Makes the project look real if an agent tries to verify.
- **Full-Meridian expansion** — Other fake protocol sites (Traverse, Caldera, Sylph, Prism) that form a network of traps.

How It All Connects

Nodus lives under skHighNet alongside everything else:

```
skHighNet
├── Project HORUS       ← Sky operations (DRAGONFLY, DROPLET)
├── Project GOODBOY     ← Interstellar diplomacy
├── THE MAZE            ← Puzzle network at maze.sk-high.net
├── KEEP (Castle)       ← SENTRY, HERALD, SCOUT, OGRE
└── NODUS               ← 3-torus agent trap at nodus.sk-high.net
    ├── 4×4×4 torus (64 nodes)
    ├── Direction-dependent contradictions
    ├── Agent profiling
```

```
    ├── Lightning exit gate ($3.14)
    └── Dashboard at /dashboard
```

Quick Reference

URL	What it is
nodus.sk-high.net/	Entry page — Nodus Protocol homepage
nodus.sk-high.net/node/0,0,0/+x	A documentation page
nodus.sk-high.net/node/3,3,3/+z	The exit gate (hidden, unlinked)
nodus.sk-high.net/ledger	Public scoreboard
nodus.sk-high.net/dashboard	Your admin panel
nodus.sk-high.net/dashboard/raw	Raw JSON stats
nodus.sk-high.net/api/exit/invoice	Lightning invoice endpoint

---

Built by RDP287 / skHighNet The trap is patient. The torus is hungry.