TIME'S UP! KEEPING TRANSPORT PROTOCOLS REAL-TIME!

INTRODUCING NANOPING

NANOPING IS A DROP-IN SOLUTION TO PROVIDE REAL-TIME CONNECTIVITY - IN ANY APPLICATION AND SERVICE.

- NanoPing operates at the network packet level
- NanoPing is a plug and play application that allows pain free integration of our dedicated protocol
- NanoPing is available as a software application for immediate evaluation and use
- Dedicated PoC in which can run in your network with your traffic

Crafting real-time solutions on raw UDP, TCP, or QUIC is notoriously complex - and often feels impossible.

NanoPing solves this challenge by blending cutting-edge innovations in protocol design, FEC, networking, scheduling, and monitoring into a seamless, powerful solution.

With NanoPing, your team can shift focus from the headaches of connectivity to unlocking the full potential of your applications / services.

REAL-TIME COMMUNICATION IS BECOMING THE NORM

TELE-DRIVING

Latency of over 200ms results in collisions

REMOTE-CONTROLLED EQUIPMENT

Packet loss and jitter can make machinery unpredictable

DRONES

Minor IP disruptions can break video, telemetry, and control

USE CASES

HOW DO YOU ENSURE REAL-TIME COMMUNICATION ACROSS UNSTABLE AND UNPREDICTABLE NETWORKS?

Across drones, remote-controlled machinery, and tele-driving, reliable real-time connectivity is critical, yet often the weakest link. Whether operating in rural areas with scarce and unstable coverage, or in urban environments with dense but congested networks, wireless connections are exposed to packet loss, jitter, variable latency, interference and sudden link drops. Even small disruptions can result in degraded video feeds, delayed telemetry, sluggish control responses, and reduced situational awareness which directly impacts safety, efficiency, and operational reliability.

While encryption protects data confidentiality, it does not address availability. Network-level issues such as congestion, interference, handover failures or jamming affect IP connectivity itself, regardless of the application or protocol in use. In real-time operations, milliseconds matter. Stable control, up-to-date video, and predictable latency require more than just network access, they demand intelligent, resilient network behavior that can adapt continuously to changing conditions.

WITH NANOPING, REAL-TIME OPERATIONS BECOME SMOOTHER, SAFER, AND MORE RELIABLE:

- **Multi-connectivity routing** lets you use Wi-Fi, cellular, and satellite together. NanoPing continuously monitors and routes data through all available links. Creating one fast reliable unified path.
- **Link prioritization** keeps costs under control without compromising performance, allowing you to define which networks are preferred when cost profiles differ.
- **Advanced error correction and network coding** remove the need for retransmissions, keeping your video feed sharp and all data flowing in true real time - even when the network performs poorly.
- **Seamless handovers** through intelligent multi-path routing keep control and video stable as devices move between networks.
- **Load balancing across links** prevents any single network from becoming a bottleneck, ensuring consistent performance.
- **Congestion prediction and avoidance** let NanoPing detect early signs of overload and steer traffic away before latency spikes appear.

NanoPing keeps real-time operations stable, responsive and reliable, even under unpredictable and challenging network conditions.

BUILDING REAL-TIME COMMUNICATION IS HARD

YOUR EXISTING STACK

Bespoke networking solutions are built on existing networking solutions (e.g. single link, TCP, etc), and implementations need to consider the existing infrastructure

YOUR USER'S UNRELIABLE NETWORK(S)

Mobile and wireless networks are a de-facto norm, but the underlying protocols have not caught up

IMPROVE POOR USER EXPERIENCE

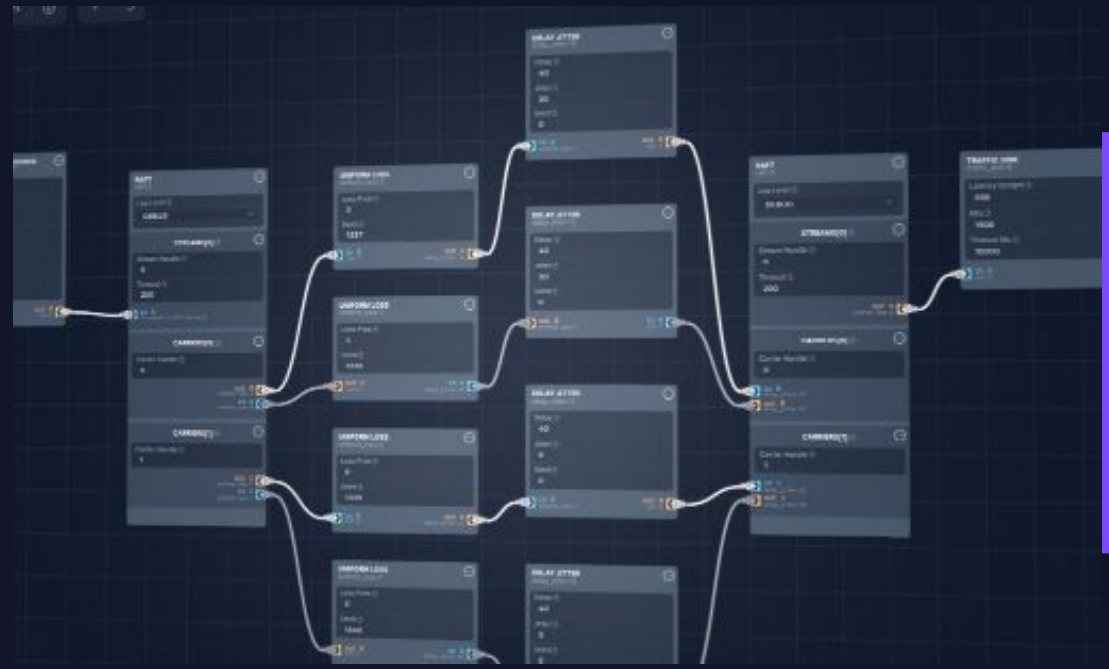
Applications usually do not have control over the network of the user, and trying to make the network itself reliable is not a viable option

MAKING UNRELIABLE NETWORKS RELIABLE

For real-time applications, the requirements are simply greater. It's your job to need to accommodate any constraint (platform, network, use case, etc...) or you're at fault

BUILDING REAL-TIME COMMUNICATION IS HARD

```
user@nanoping sudo np up
Started. Serving dashboard on http://127.0.0.1:8081
```



THE FUNDAMENTALS

RLNC

Erasure coding ensures efficient recovery and utilization of multi-path

MONITORING

Extract and monitor important KPIs to ensure value delivered and to spot issues

INTEGRATION

Binary or software integration in end-user applications / services

LATENCY AWARE TRANSPORT PROTOCOL

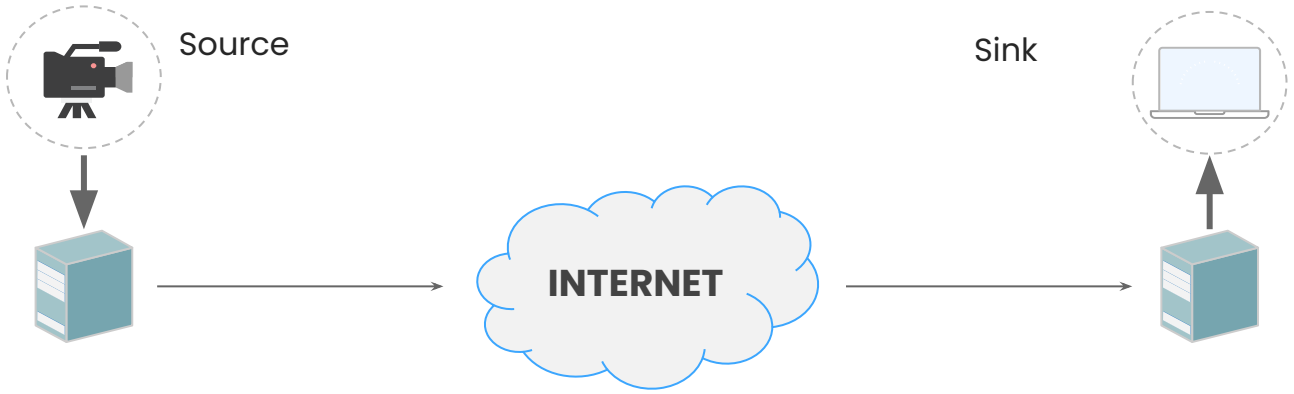
Tracking latency and reliability across multiple network paths (if available) - dynamically adjusting strategy to ensure reliable delivery

CUSTOMIZABILITY

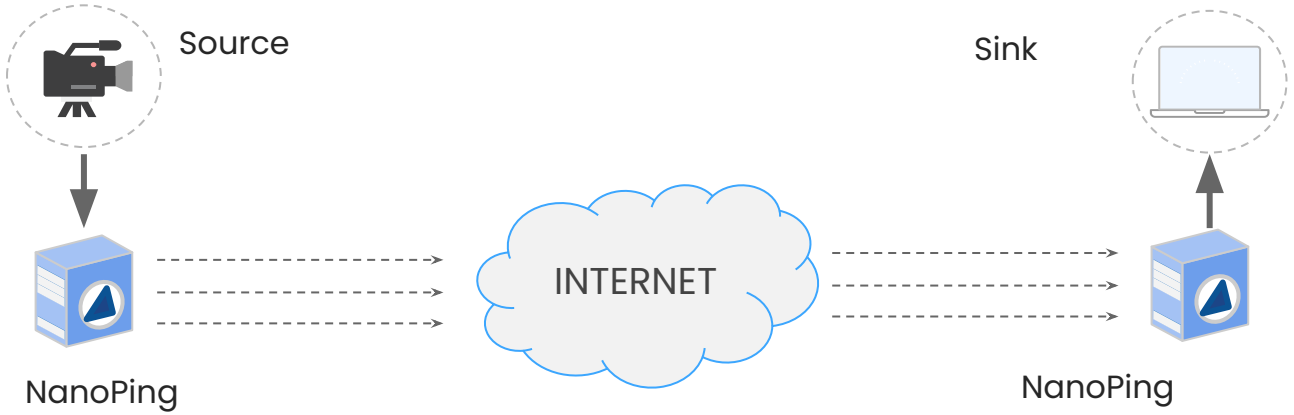
Adapt to different application requirements

NANOPING DATA-PLANE

TODAY



NANOPING



TCP/UDP TRANSPORT

- Not latency aware
- Not multi-path
- Only retransmissions
- No observability
- No stream prioritization
- No application feedback

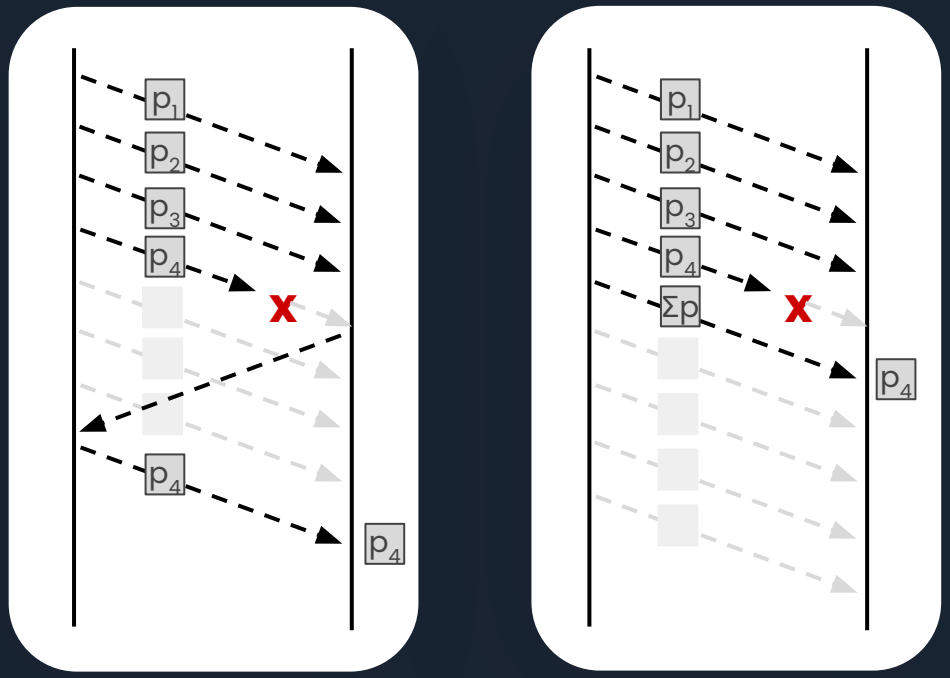
TCP/UDP TRANSPORT

NANOPING TRANSPORT

+ Latency aware
+ Multipath
+ Adaptive packet FEC/ARQ
+ Built-in observability
+ Stream priorities
+ Application feedback

TECH DEEP DIVE

One of the key reasons for using **ECC/FEC** is to **minimize per-packet delay** for latency sensitive applications.

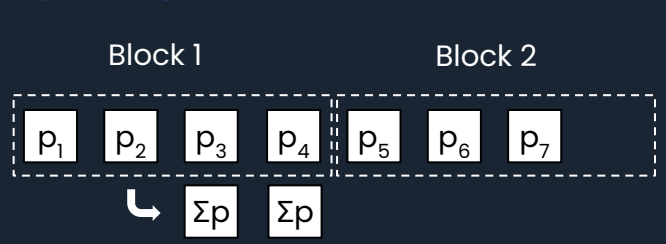


ARQ RECOVERY LATENCY:
1 RTT per retransmission

ECC/FEC RECOVERY LATENCY:
Distance to the repair packet

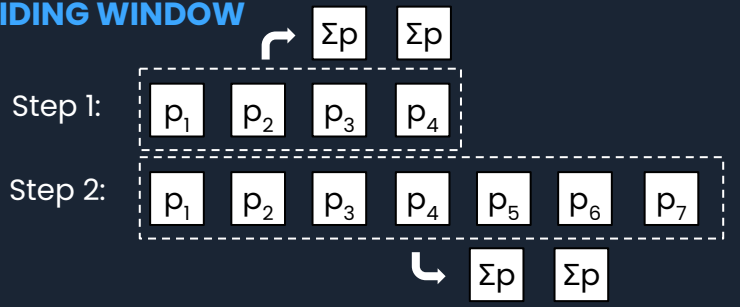
- By interleaving repair packets ECC/FEC based schemes can provide repair for packet loss faster than retransmissions.
- Picking the right algorithm is crucial!
  Fundamentally two types of ECC/FEC schemes exist:
  1. Block based (e.g. Reed Solomon)
  2. Sliding window based

BLOCK BASED



- Repair only possible at end of block
- Repair only covers packets in one block
- Time to repair dictated by block size

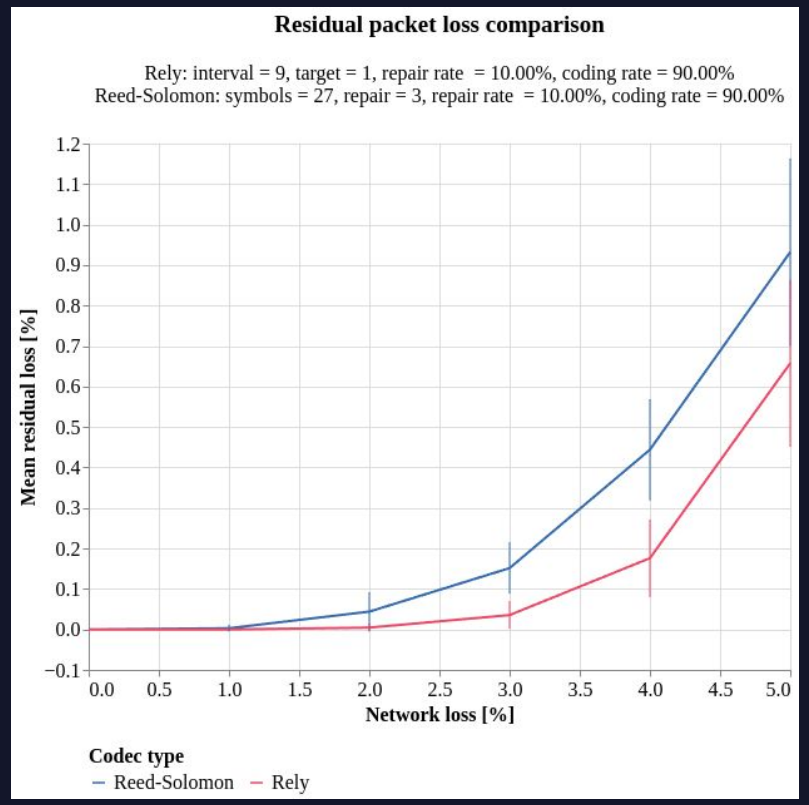
SLIDING WINDOW



- Repair can be generated at any time to ensure proper protection
- Repair can cover any packet in window

COMPARISON

Sliding window coding provides better or equal performance for (1) packet loss protection (2) per packet delay:



IMPLEMENTATION BENEFITS

- Using block ECC/FEC all repair and computations is performed at the end of the block - causing computational and bandwidth spikes
- Using sliding window repair can be evenly distributed providing a smooth computations and bandwidth usage
- Using block ECC/FEC requires management of multiple encoders and decoders per stream
- Sliding window codes naturally fit real-time streaming / packet-processing use-cases - with a single encoder/decoder per stream
- Using block ECC/FEC low-throughput applications require small blocks (reducing packet loss protection)
- Sliding window codes are elastic and seamlessly adapts to the traffic pattern (from low to high throughput)
- Using block ECC/FEC low-throughput applications require small blocks (reducing packet loss protection)
- Sliding window codes require no fragmentation of input data

nanoping