Pacer: Comprehensive Network Side-Channel Mitigation in the Cloud

Aastha Mehta, Mohamed Alzayat, Roberta De Viti,
Björn B. Brandenburg, Peter Druschel, Deepak Garg

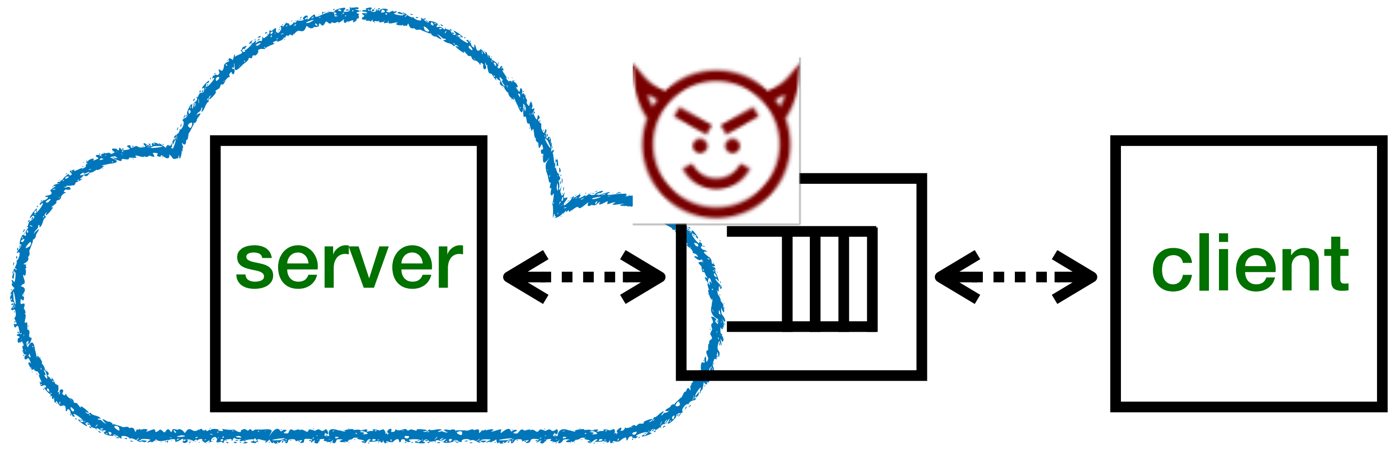
MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

THE UNIVERSITY OF BRITISH COLUMBIA

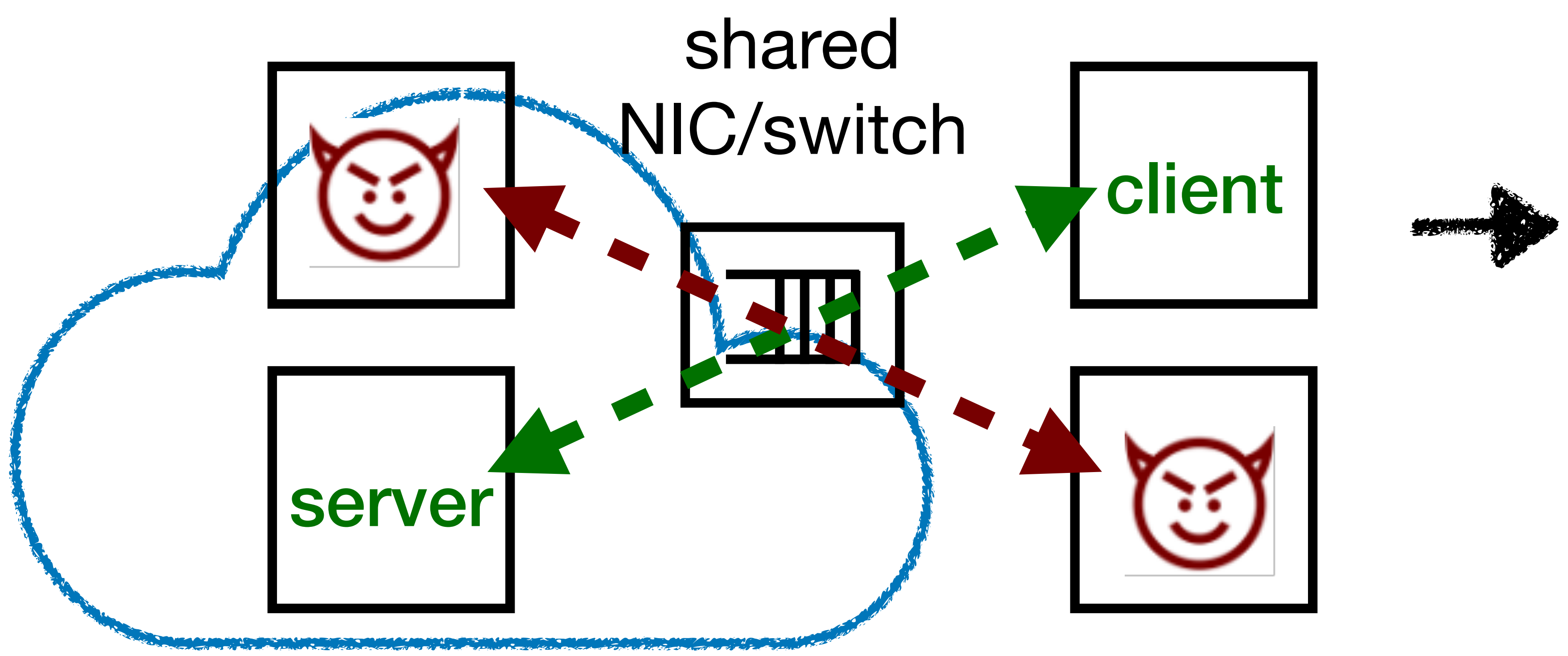
Network side channels

Step 1: Observing victim's traffic shape

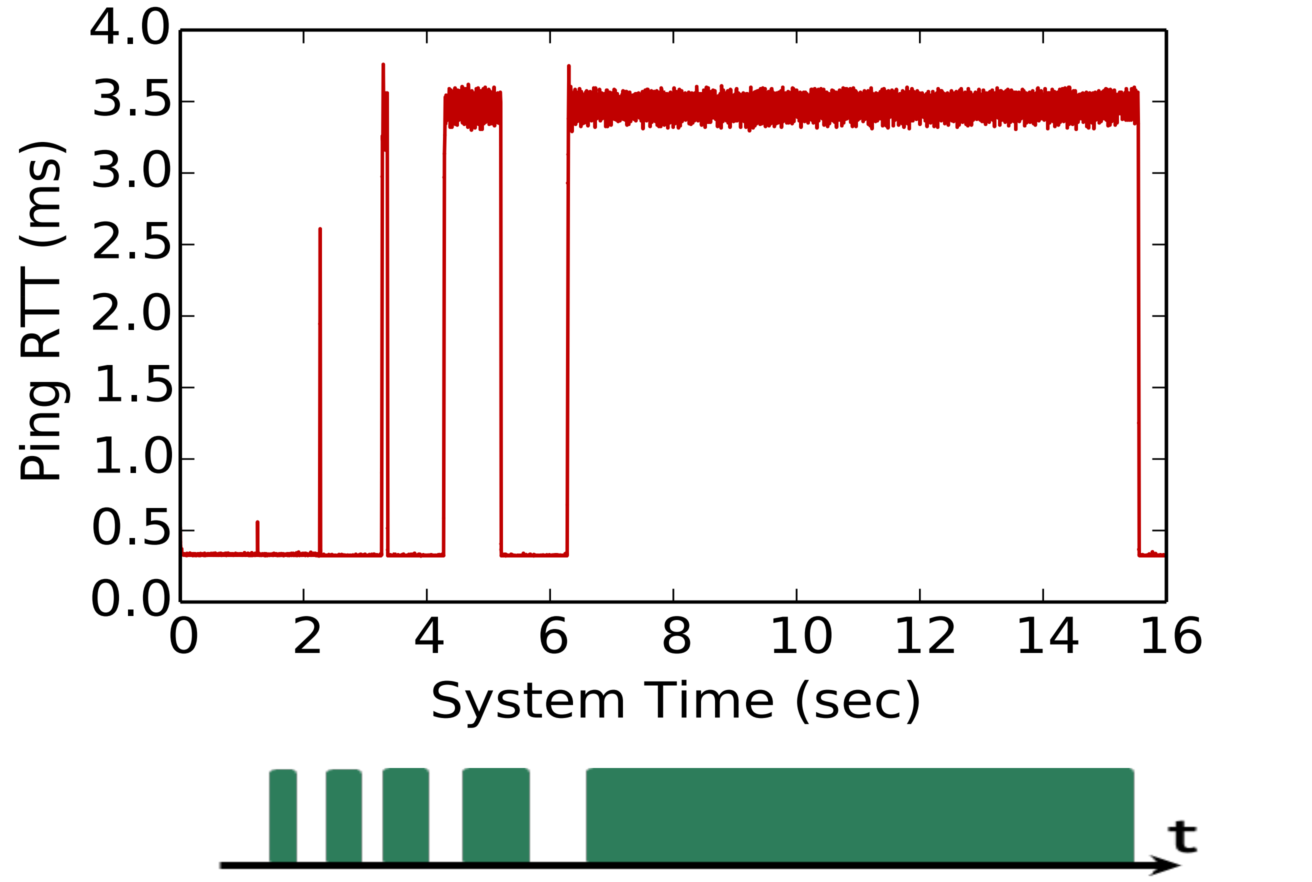
A. Direct observation at a link



B. Contention at a shared link



Adversary's timing measurements



Adversary can infer victim's traffic shape (burst sizes, timing)

Step 2: Inferences from traffic shape

- Traffic content (web pages[PETS02], VoIP[S&P08], videos[Security17])
- Users' medical, financial secrets[S&P10]

Pacer: Key ideas

Cloaked tunnel abstraction ensures secret-independent traffic shape by design

Paravirtualized implementation for IaaS clouds

Batched transmission scheduling masks architectural side channels

In the paper:

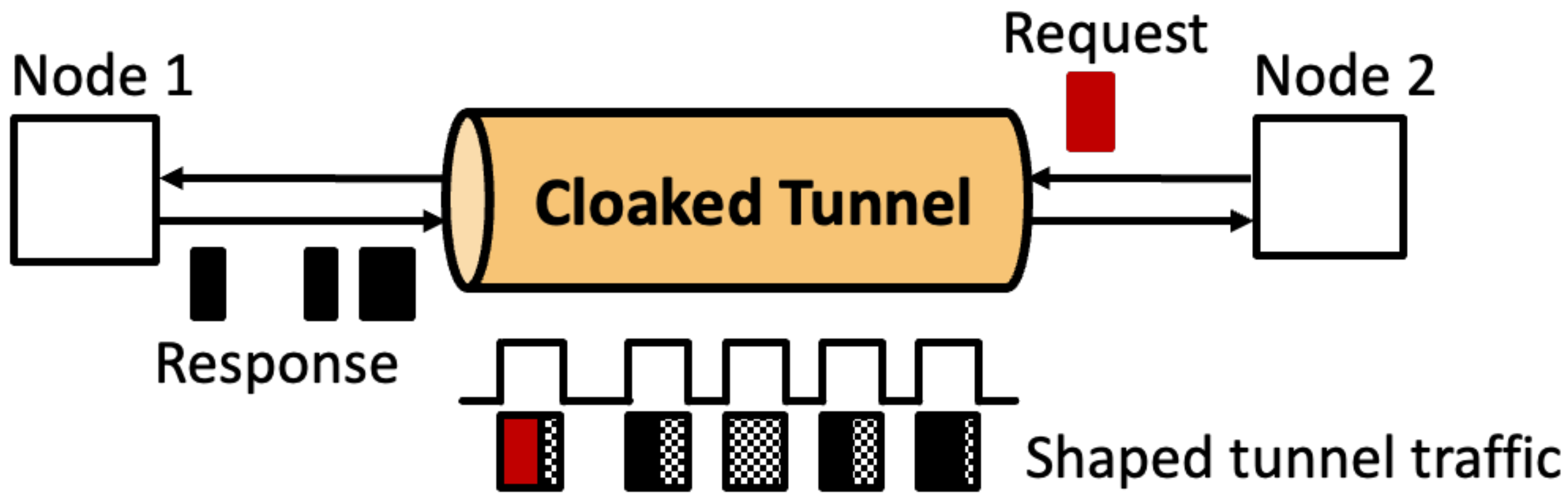
Exploiting public inputs to efficiently shape bursty traffic

Graybox profiling to automatically compute efficient traffic shapes

Detailed evaluation results

Formal proof of security

Key abstraction



Key property: Shape of the tunnel traffic is independent of secrets

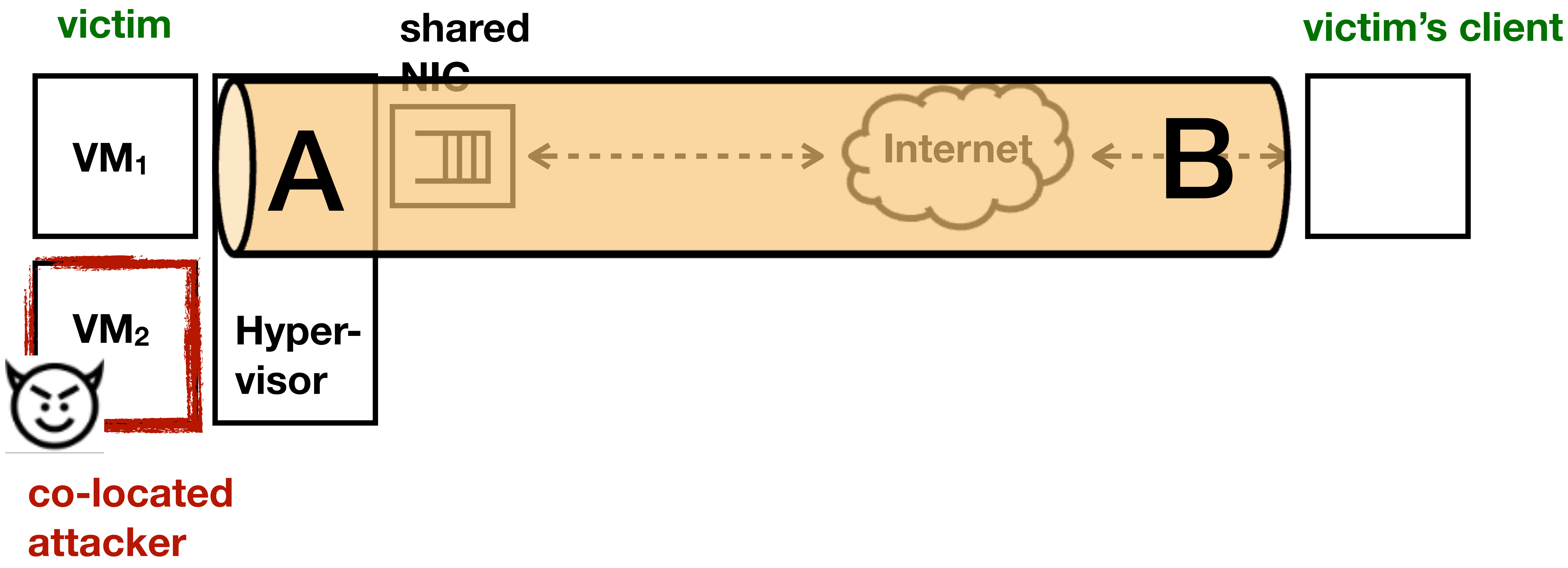
Unobservable payload

- Hide flow control
- Encrypt all packets to hide padding
- Elicit same response for padding and payload

Secret-independent timing

- Transmit on schedule
- Must not be delayed by secrets

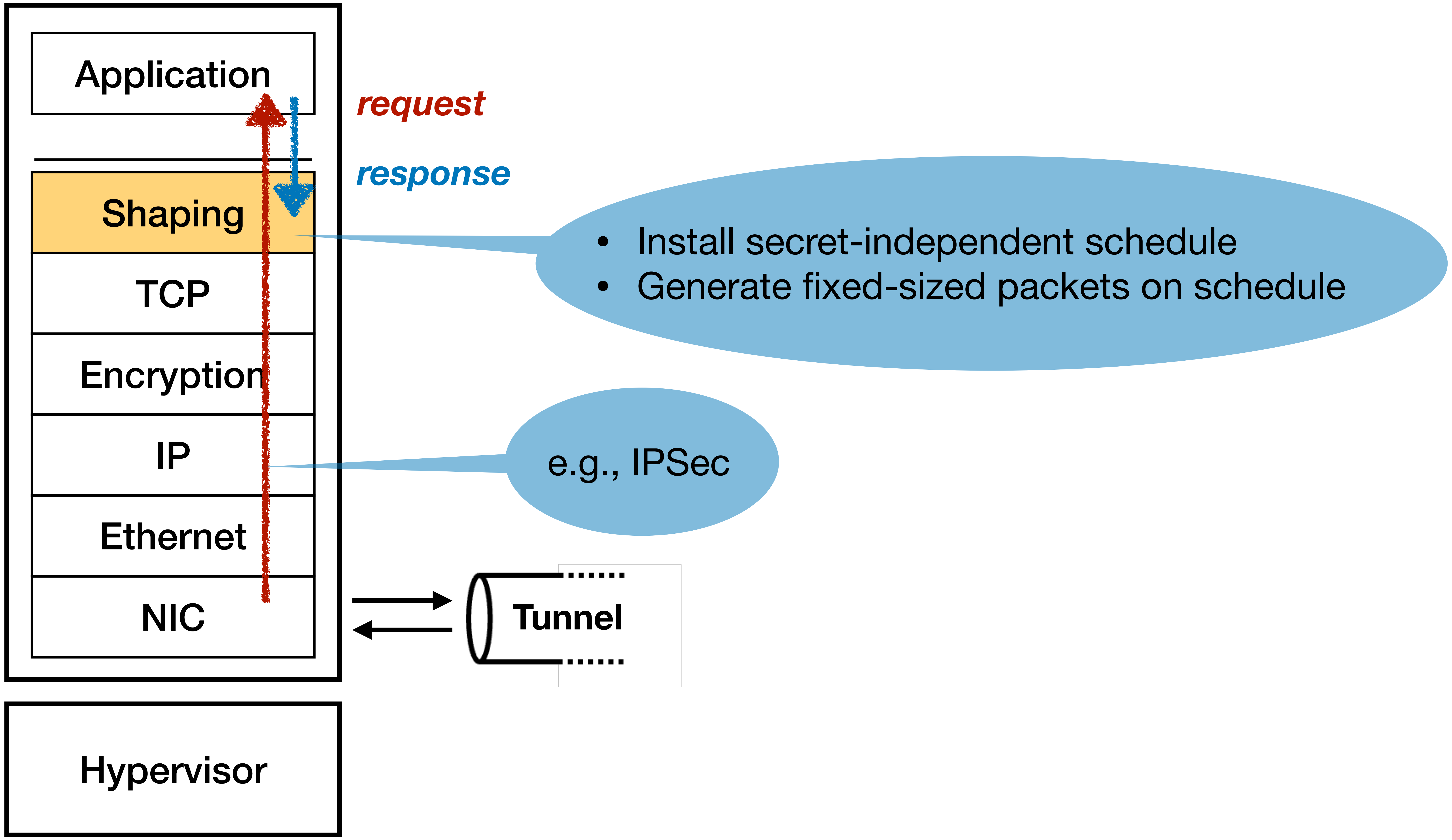
Tunnel endpoints for Cloud tenants



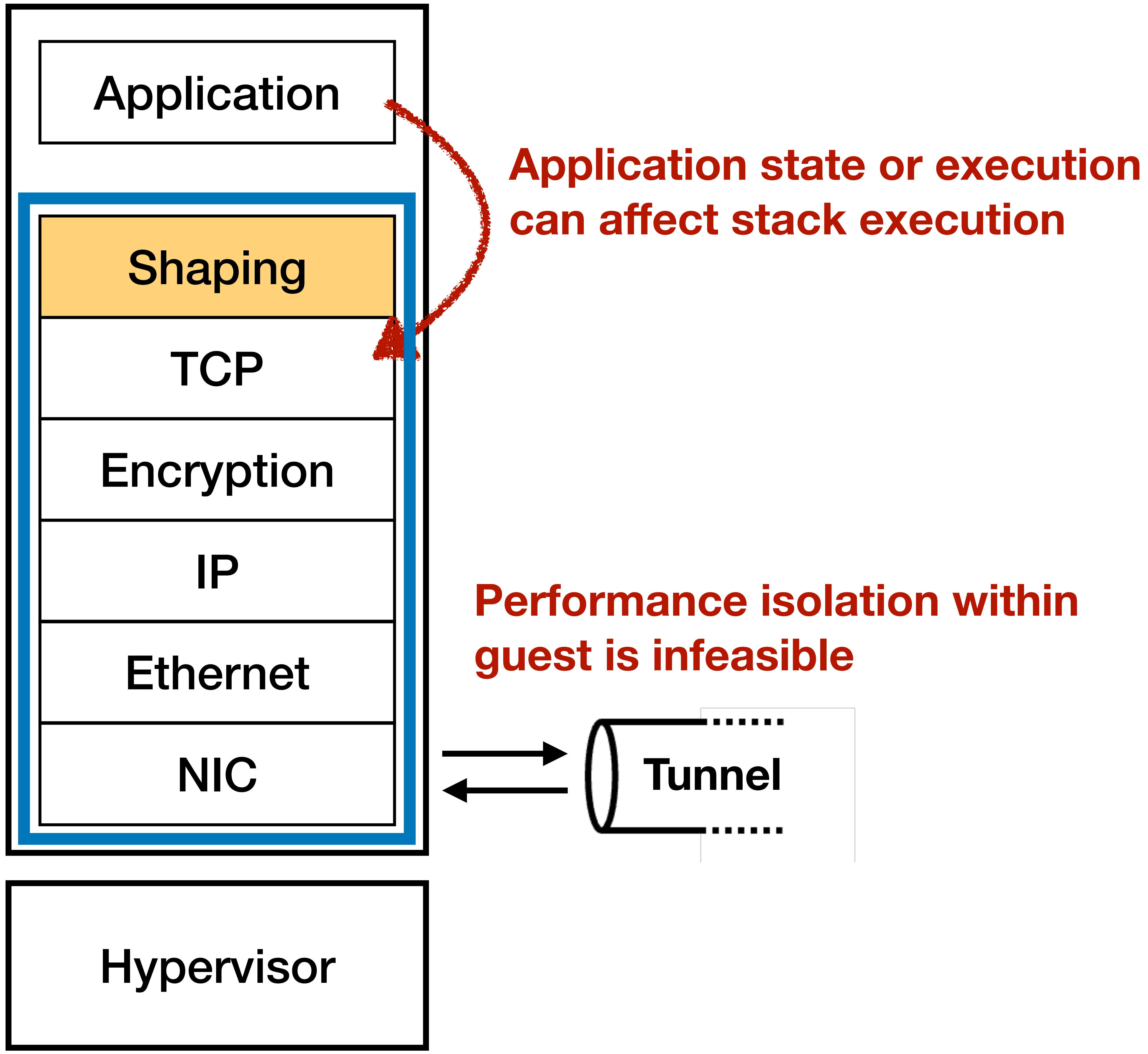
Endpoint A integrated with VM host

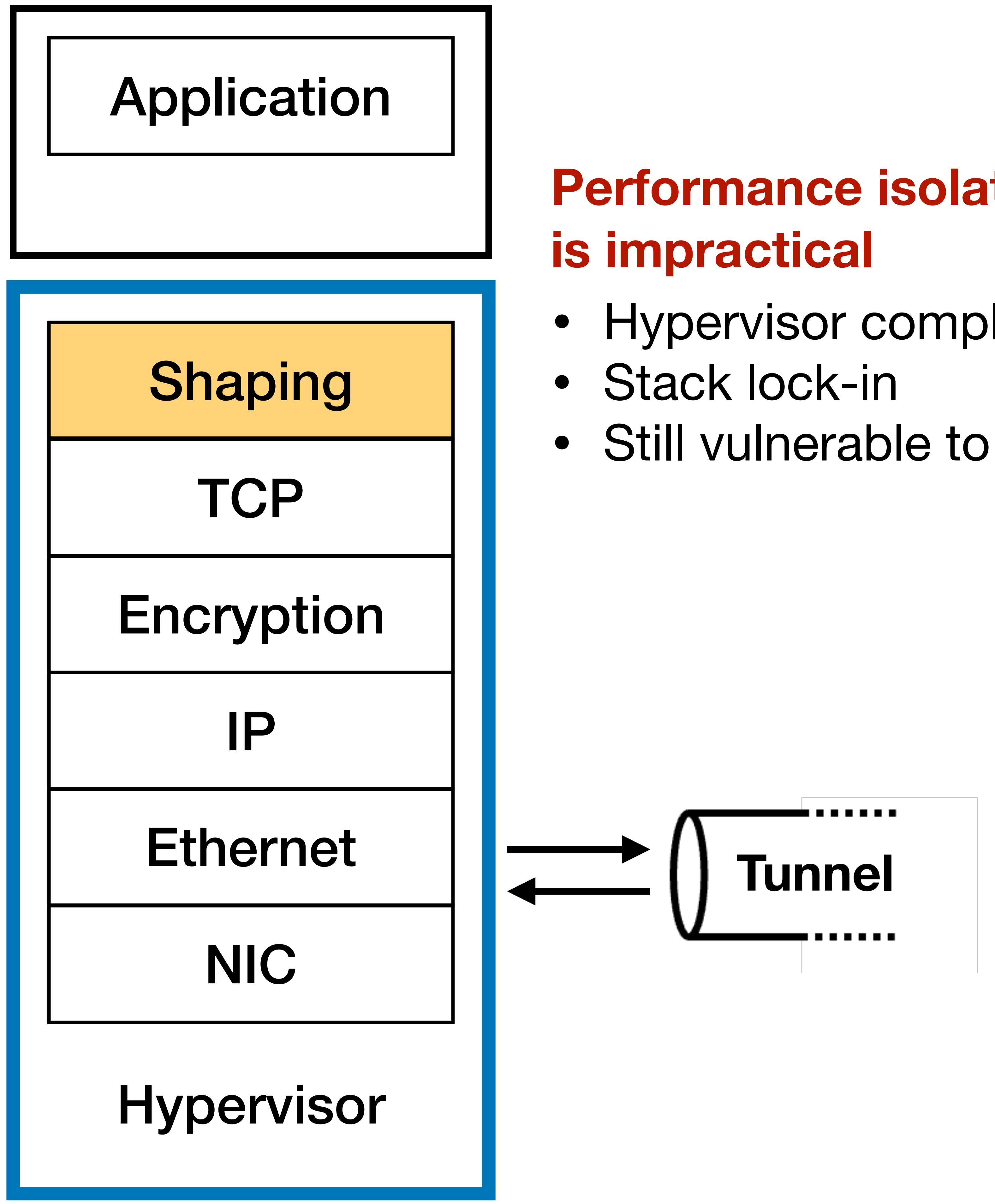
Endpoint B on client device or premises

Endpoint architecture (conceptual)



Challenge: timing delays

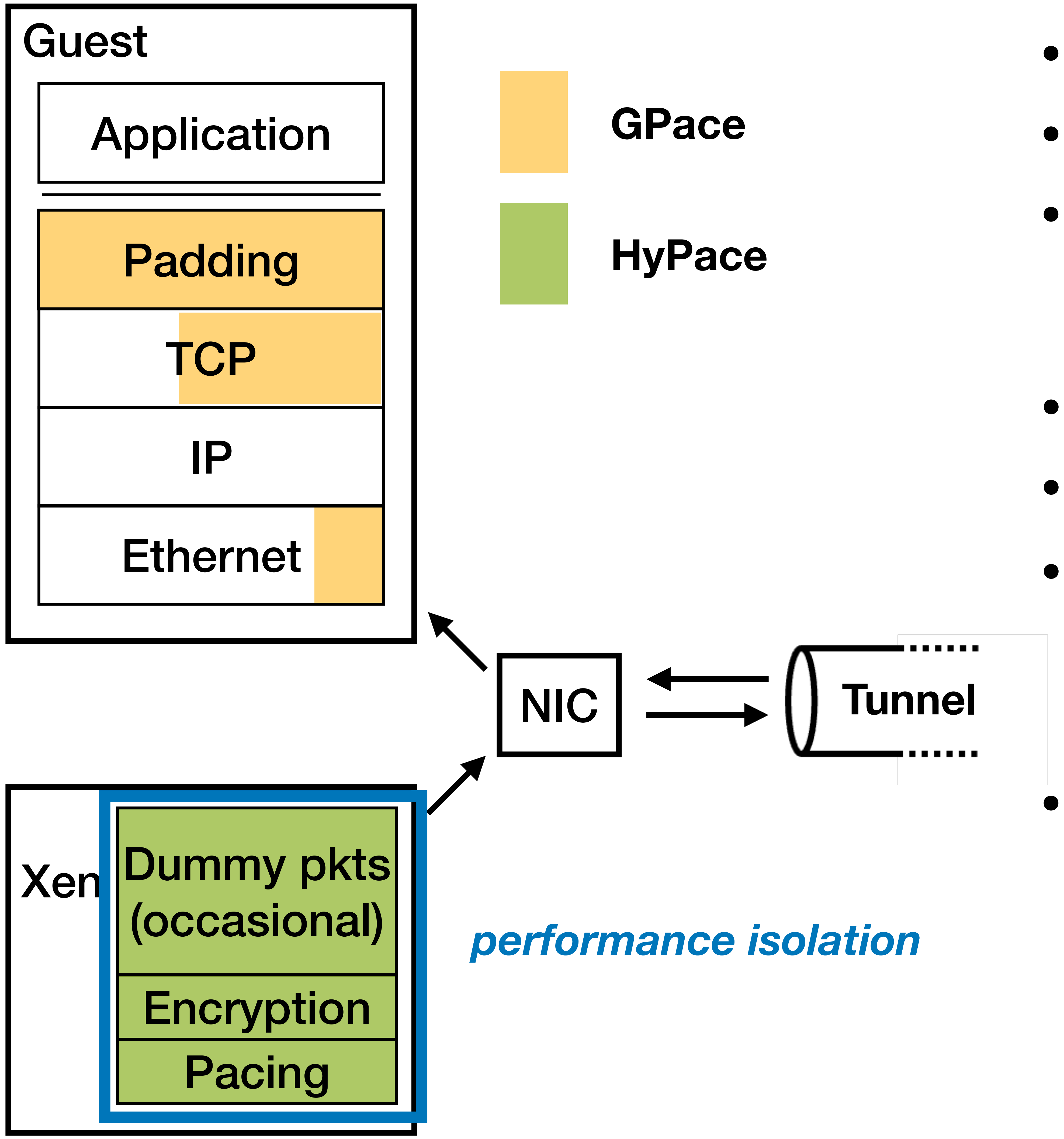




Performance isolation in HV is impractical

- Hypervisor complexity
- Stack lock-in
- Still vulnerable to side channels

Pacer's cloaked tunnel



Unobservable payload

- End-to-end padding
- All packets with valid TCP seq#
- Encryption after padding

Secret-independent timing

- Transmit packets on a schedule
- Mask secret-dependent delays in HyPace
- Transmit fixed batch of packets in each interval

- Respects congestion control (practical requirement)

Evaluation

Security:

- Accuracy of CNN attack classifier
- Formal model and proof of HyPace

Performance:

- Bandwidth vs privacy overheads for static content
- Client latency, throughput of a medical website
- Streaming experience from a multi-tier video service

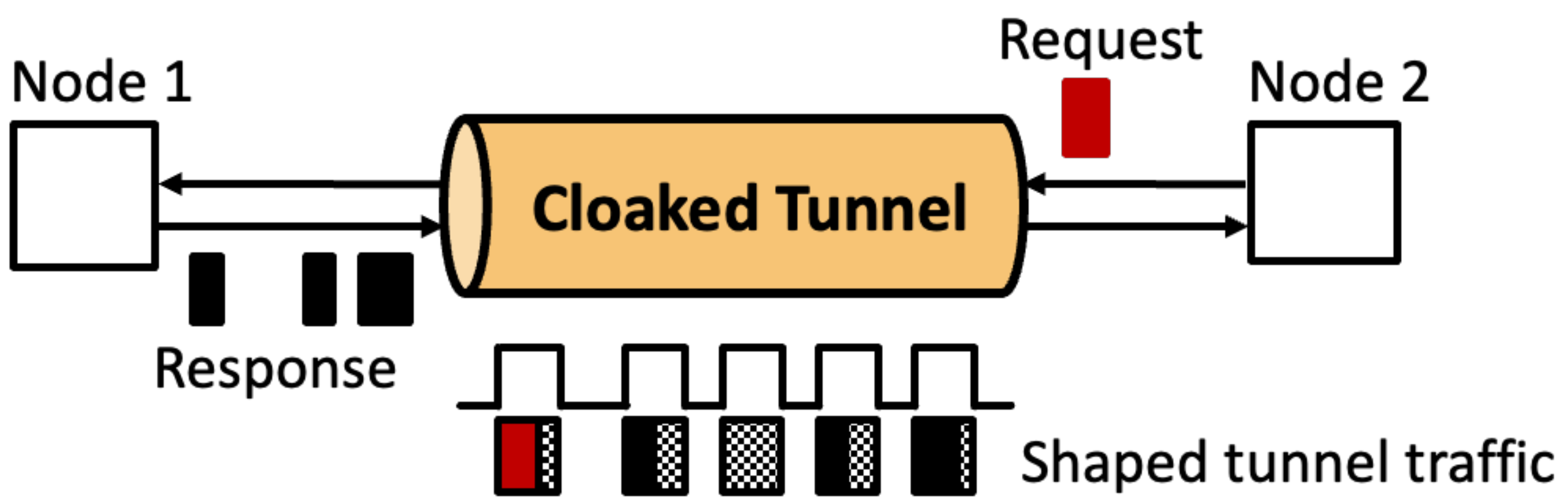
Application overheads

Metric/Application	Medical	Video
Average bandwidth overhead	143%	4x
Latency overhead	10x – 18x	1-tier: 30x 2-tier: 400x
Peak throughput overhead	14.4%	NA

No jitter at client-side

Overheads depend on application's data, workloads, and public inputs

Summary



✓ **Secure by design**
Secret-independent traffic shapes
Masking secret-dependent delays

✓ **Efficient and practical**
Public input-dependent shaping
React to congestion control

✓ **Usable**
Minimal changes to application
Modest changes to hypervisor and guest OS

Code:
https://gitlab.mpi-sws.org/pacer