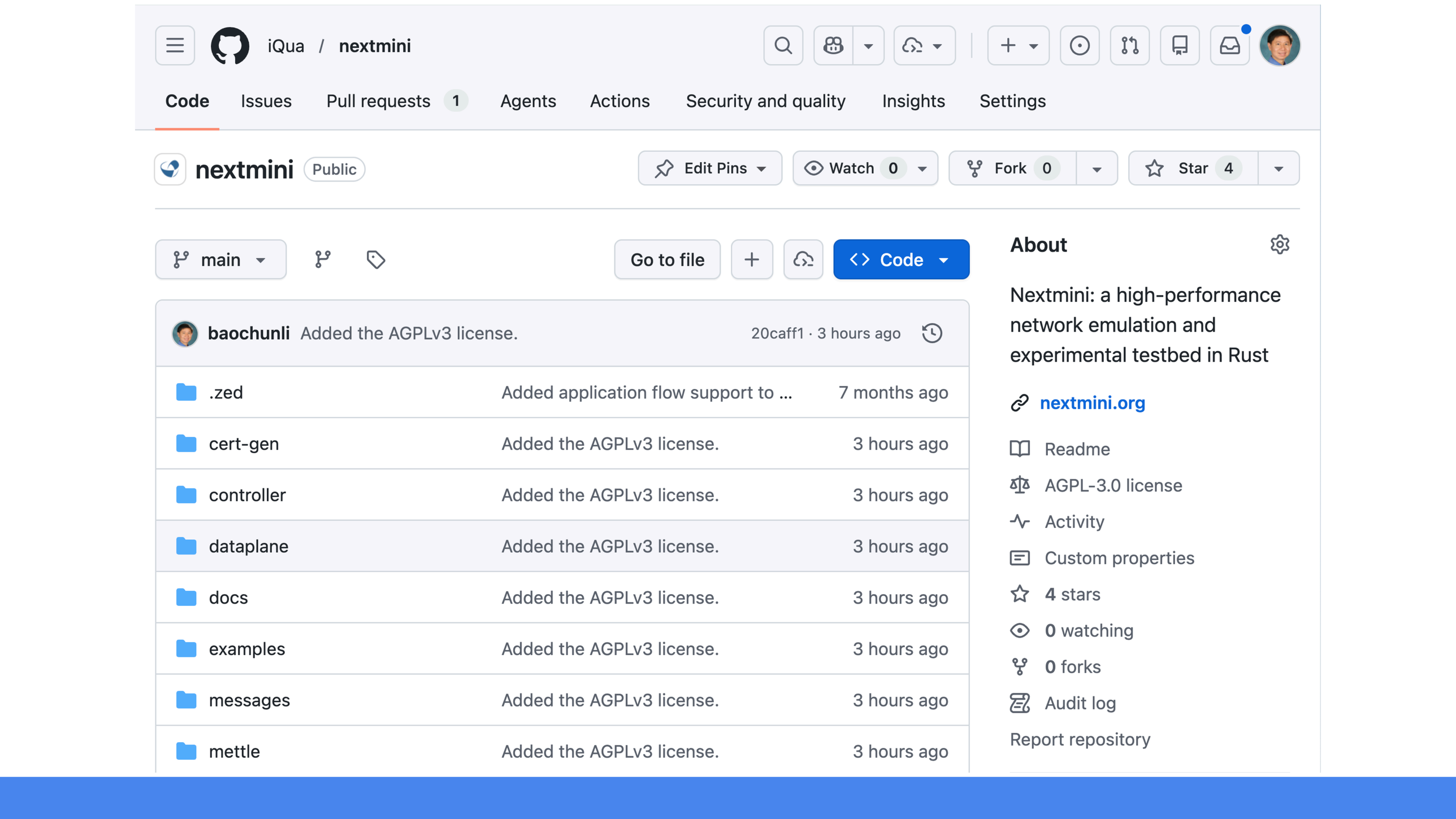
Nextmini: A New Research Testbed for Network Emulation and Experimentation

Xindan Zhang, Shengwen Chang, Baochun Li
Department of Electrical and Computer Engineering
University of Toronto


iQua / nextmini
Code
Issues
Pull requests 1
Agents
Actions
Security and quality
Insights
Settings
nextmini
Public
Edit Pins
Watch 0
Fork 0
Star 4
main
Go to file
Code
baochunli Added the AGPLv3 license.
20caff1 · 3 hours ago
.zed
Added application flow support to ...
7 months ago
cert-gen
Added the AGPLv3 license.
3 hours ago
controller
Added the AGPLv3 license.
3 hours ago
dataplane
Added the AGPLv3 license.
3 hours ago
docs
Added the AGPLv3 license.
3 hours ago
examples
Added the AGPLv3 license.
3 hours ago
messages
Added the AGPLv3 license.
3 hours ago
mettle
Added the AGPLv3 license.
3 hours ago
About
Nextmini: a high-performance network emulation and experimental testbed in Rust
nextmini.org
Readme
AGPL-3.0 license
Activity
Custom properties
4 stars
0 watching
0 forks
Audit log
Report repository

Powered by async Rust: fast!

Flexible: user-space switch design makes it easy to implement new algorithms

Underlay: supports arbitrary workloads

Both emulation and experimentation: design inspired by best practices such as containerization and orchestration

Simplicity: around 13K LOC, and users only need to supply a configuration file

Technological Foundation

Mininet's heritage

Follows software-defined networking principles with separation of concerns

Supports emulating thousands of nodes within the same physical machine by virtualizing network namespaces only

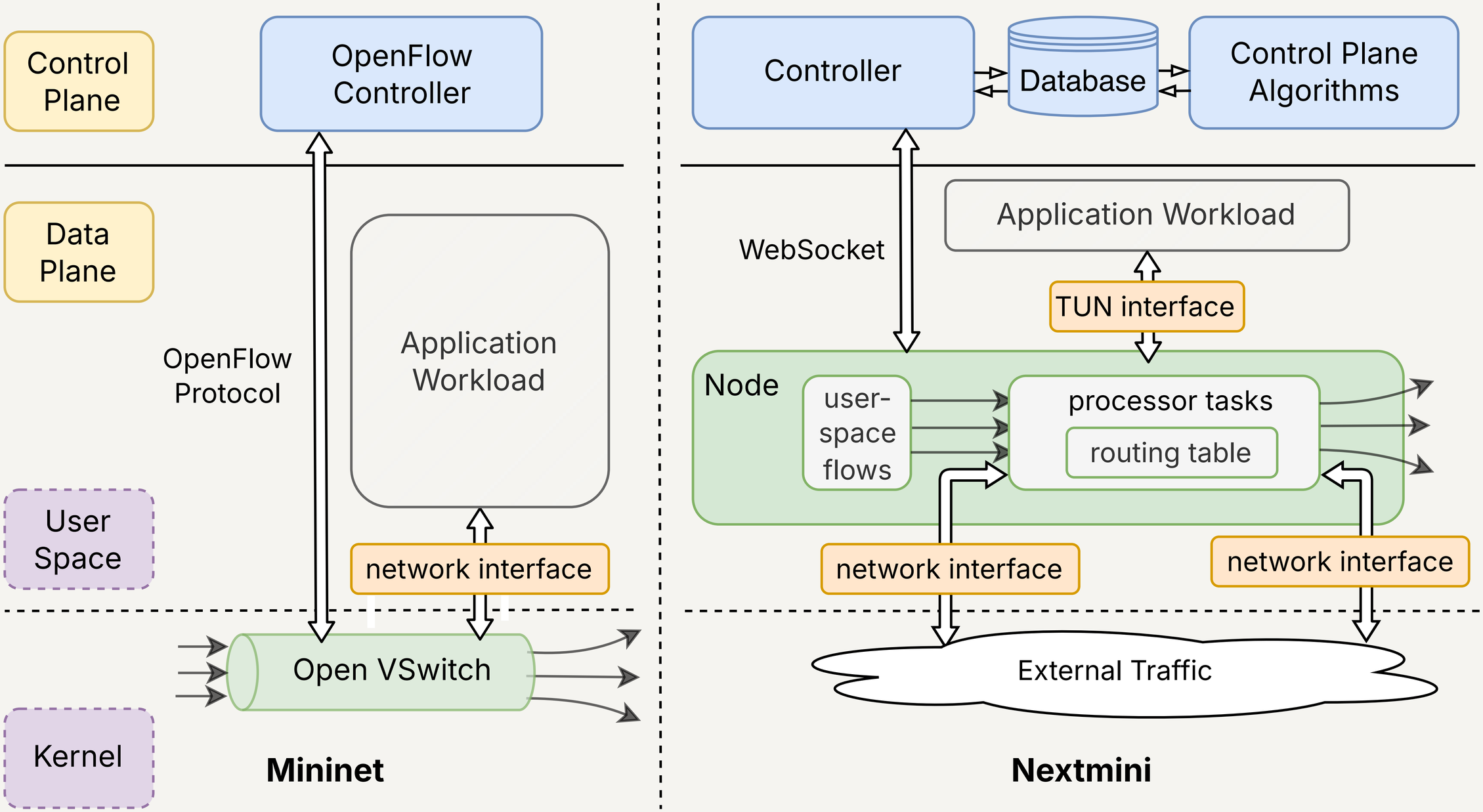
Rust's stackless coroutines

Technological Innovation

Mininet's packet forwarding: in the **kernel**

Nextmini's packet forwarding: in the **user space**

- Slower packet forwarding, but
- Offers the best possible flexibility of implementing new algorithms

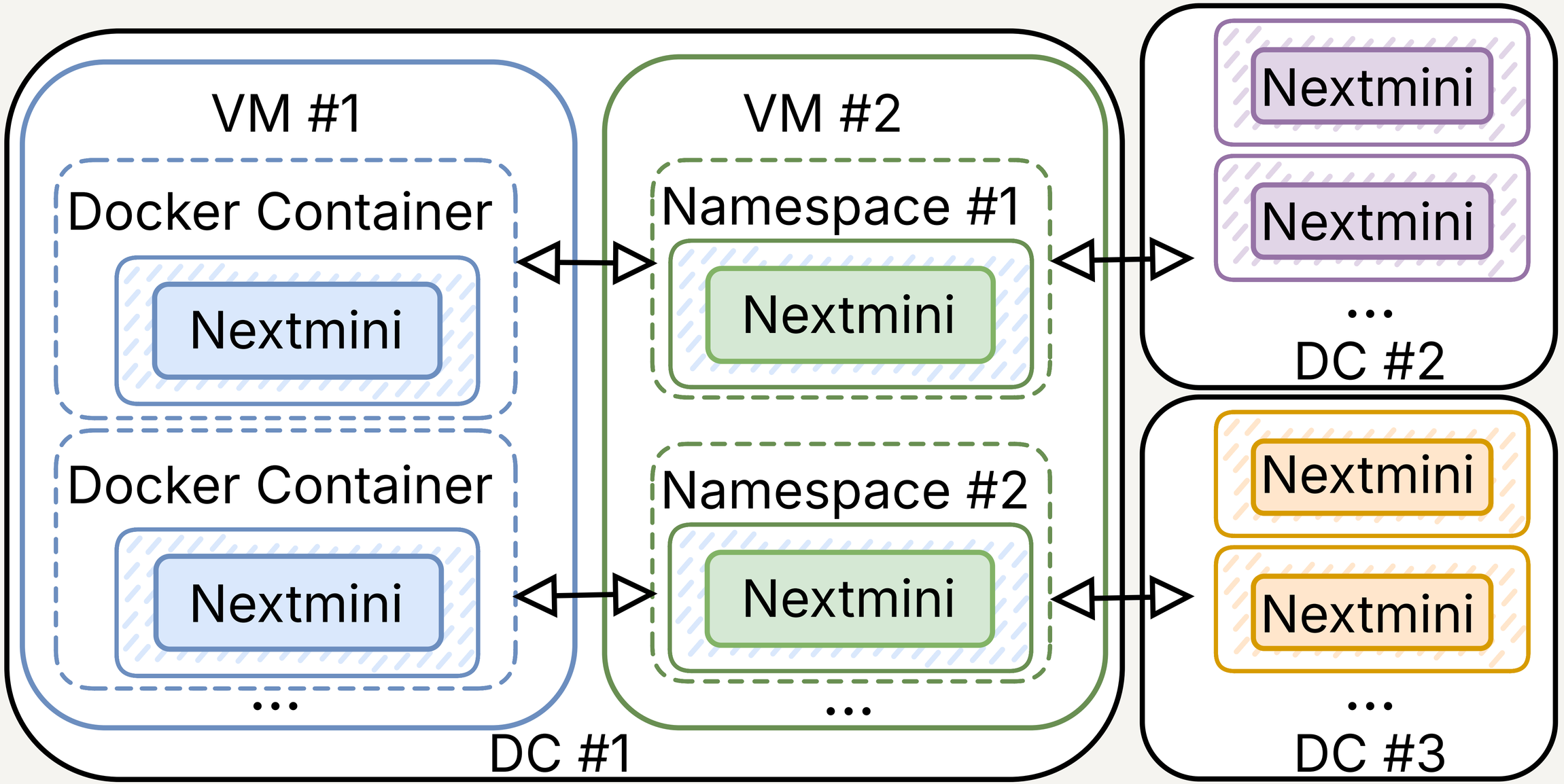


Comparing **Mininet** with **Nextmini**: a birds-eye view

Containerization and Orchestration

Docker containers: runs arbitrary application workloads in our emulation testbed

Docker orchestration: easily span multiple physical machines in the same cluster or geographically distributed globally



Nextmini with container orchestration

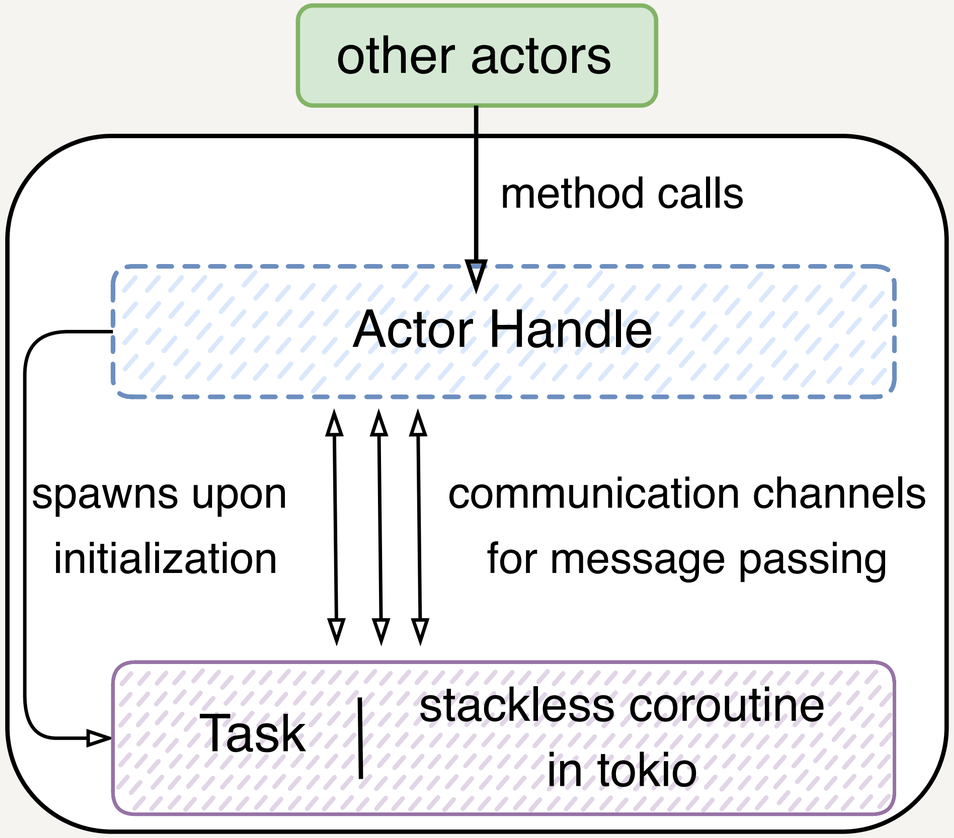
Control Plane: Rust-Powered

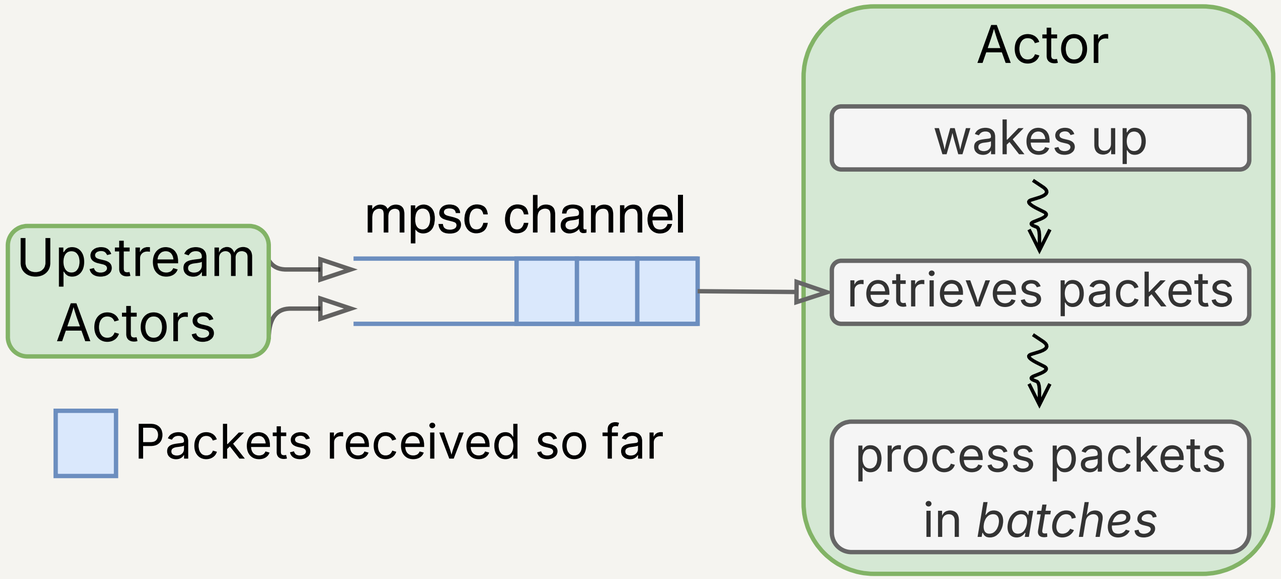
Asynchronous Rust: with the tokio executor

Two-way communication with the dataplane: in persistent WebSocket connections

PostgreSQL database and real-time triggers: stores all its states in a PostgreSQL database

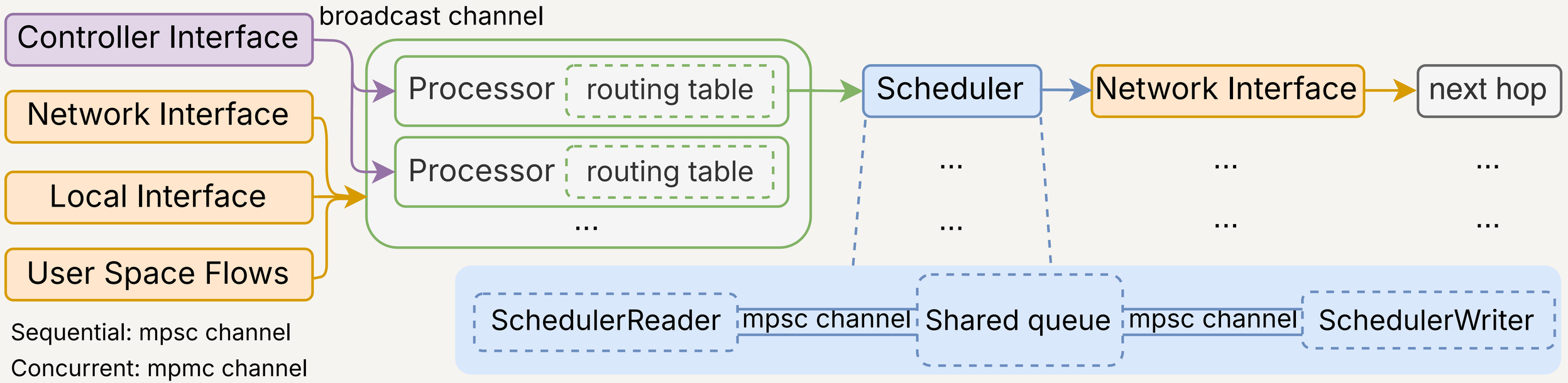
Data Plane: The Actor Model





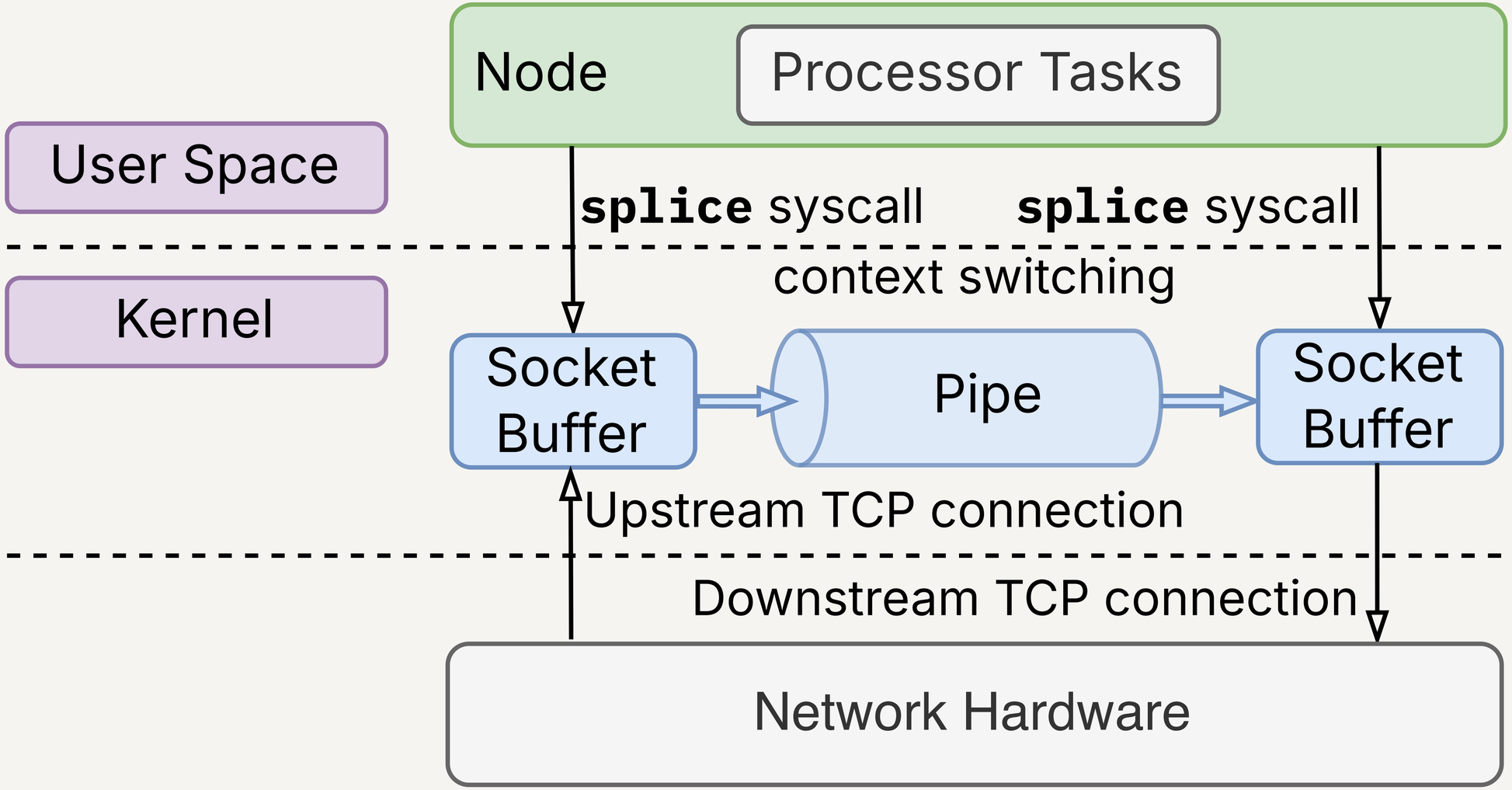
(a) The actor model: each actor is divided into a task and a handle.

(b) Processing all packets in the channel, in batches, upon resuming execution.

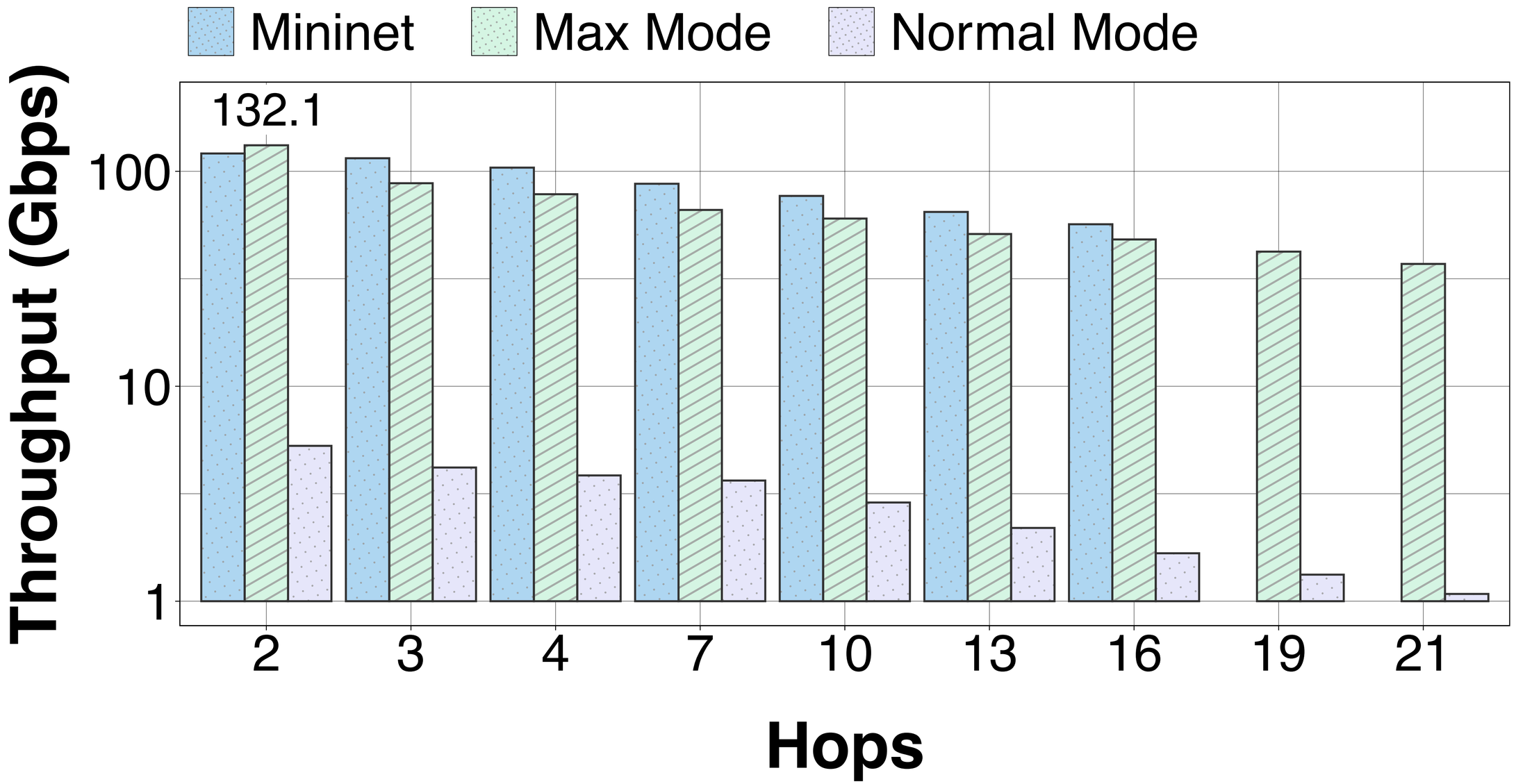


Designing **Nextmini**: a first-cut starting point.

"Have your cake and eat it too" — When performance is needed, uses the splice system call in Linux to bridge TCP connections in the kernel directly

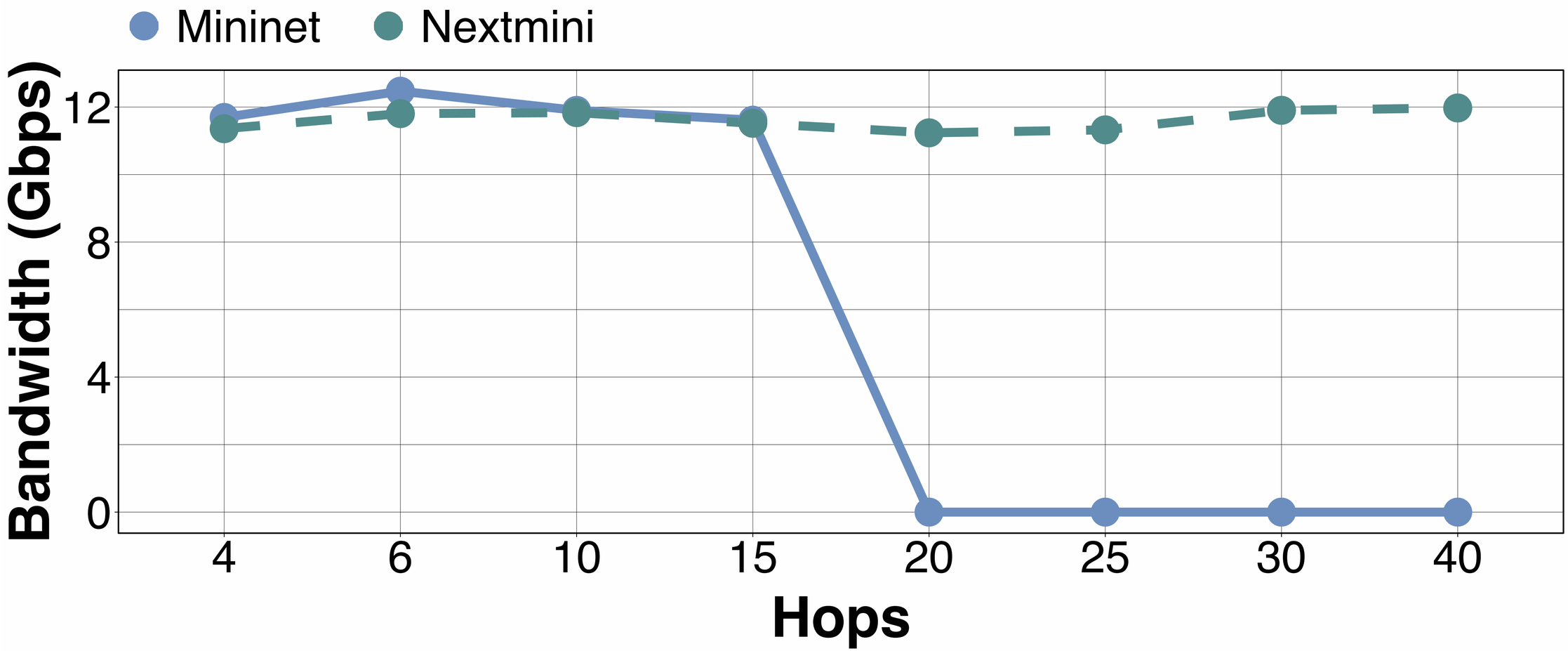


(c) Implementing Nextmini's max mode using the splice system call in Linux.



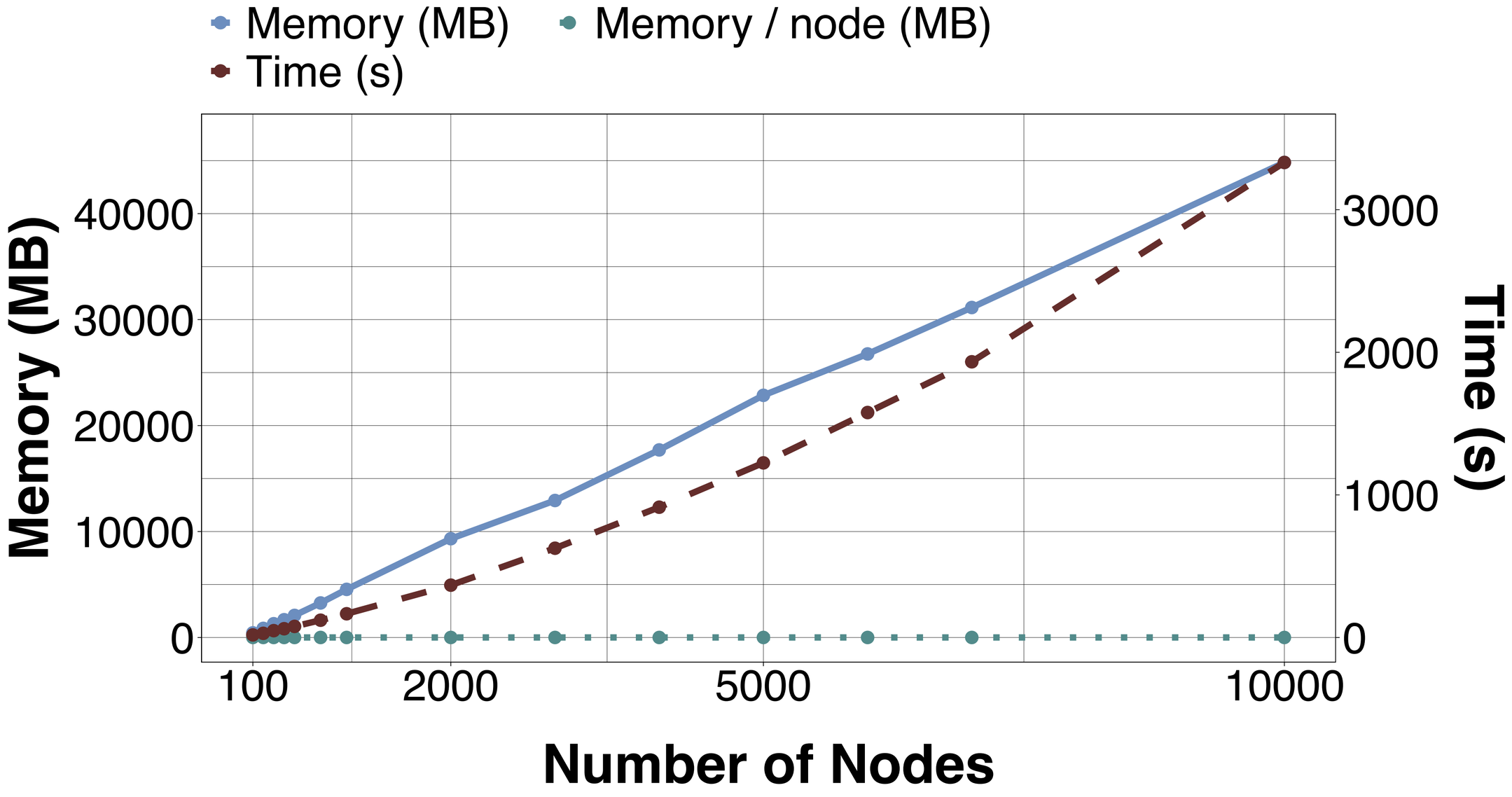
Bypassing the user space using `splice`.

Supports real-world network traffic



Achieved throughput as measured by the `curl` client.

Scalability on the same machine by virtualizing network namespaces only



Memory footprint as we scale up the number of dataplane nodes in **Nextmini**'s namespace mode

As a new emulation and experimentation testbed, we are still evolving it

https://nextmini.org