

Rethinking RPC Communication for Microservices-based Applications

Xiangfeng Zhu Yang Zhou Yuyao Wang Xiangyu Gao

Arvind Krishnamurthy Sam Kumar Ratul Mahajan Danyang Zhuo



PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

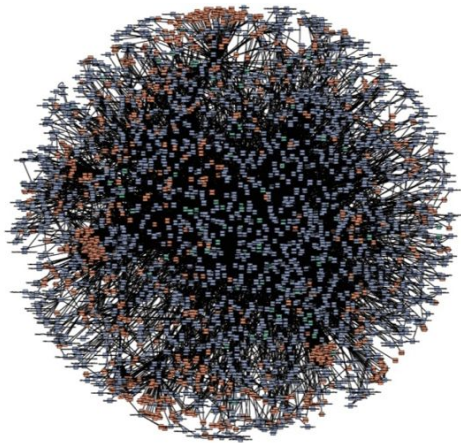
UCLA Samueli
Computer Science

Duke
TRINITY COLLEGE OF
ARTS AND SCIENCES

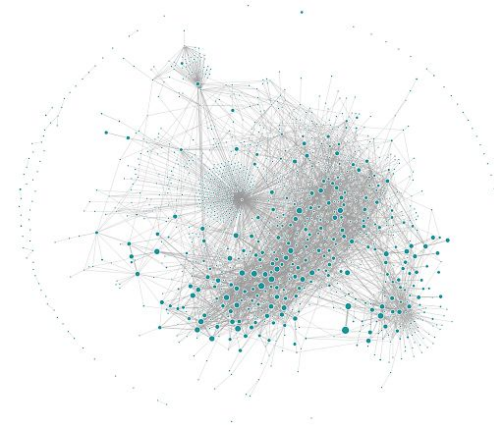
UC DAVIS
COMPUTER SCIENCE



NETFLIX

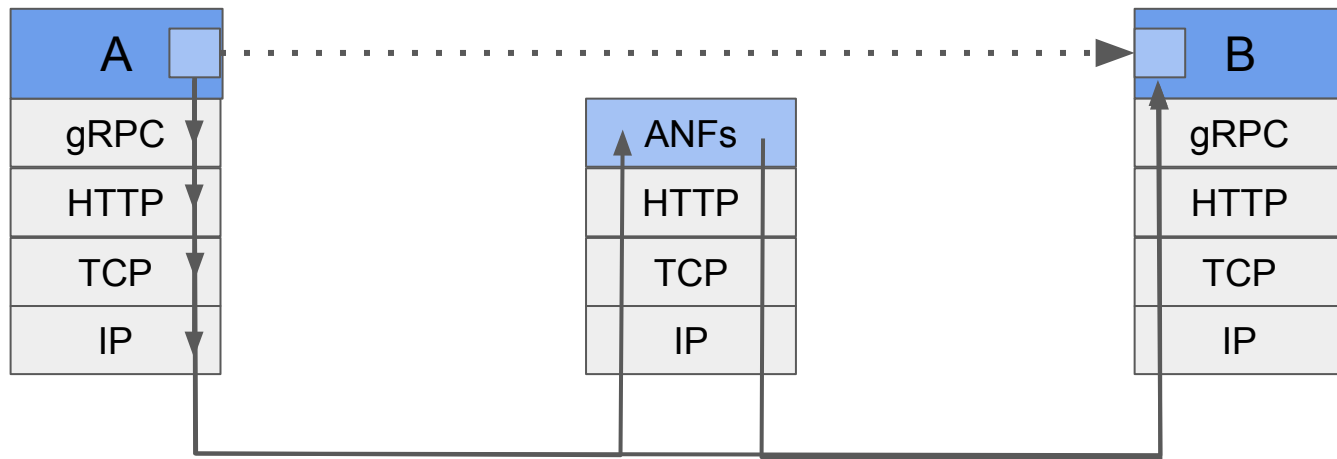


amazon

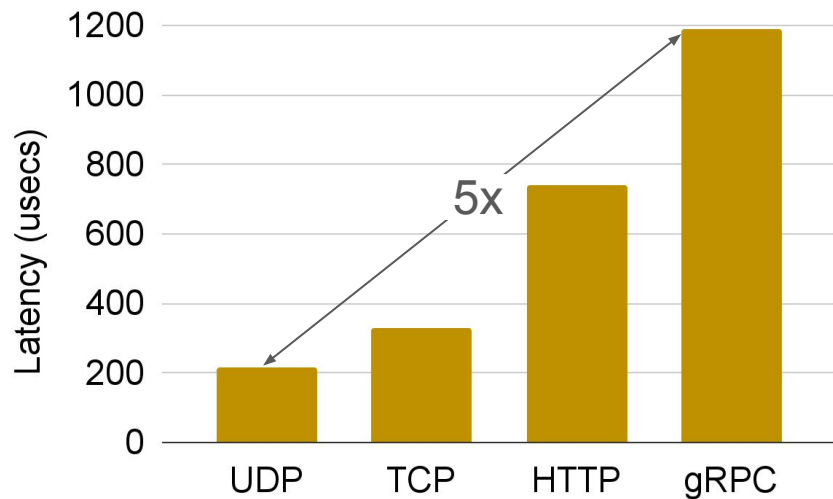


Uber

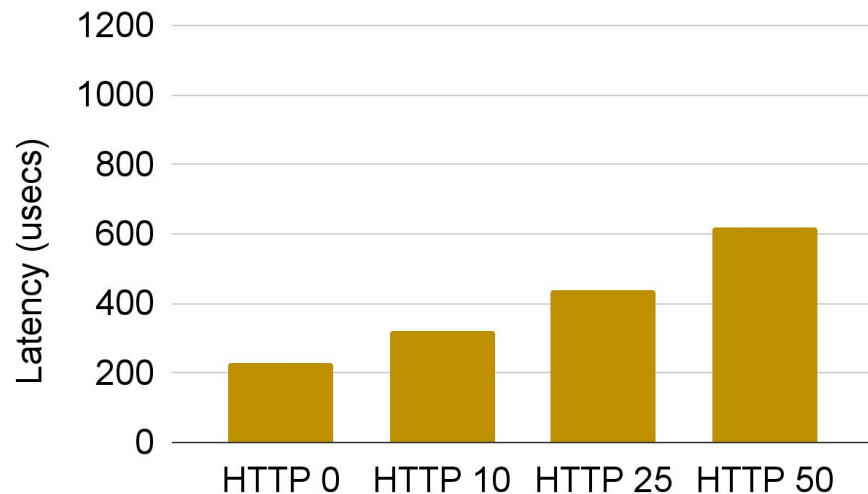
RPC communication today



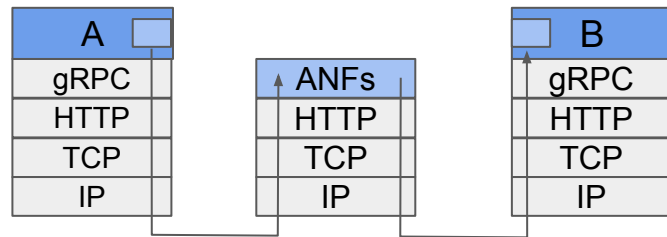
RPC communication tax



RPC protocol (w/o ANFs)



ANF processing (Envoy)

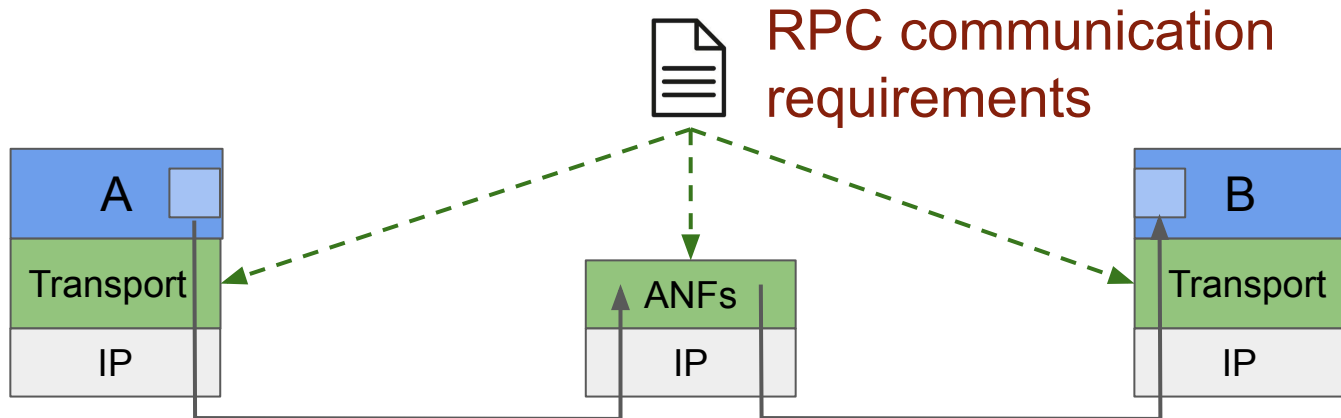


The opportunity

End-to-end path is controlled by the same entity

⇒ Don't need a stack of standard protocols

⇒ Can tightly couple the host stack and ANFs



Key research questions

How to specify RPC communication requirements?

How can we enable efficient ANF execution?

How to coordinate host stack and ANF processing?

RPC schema

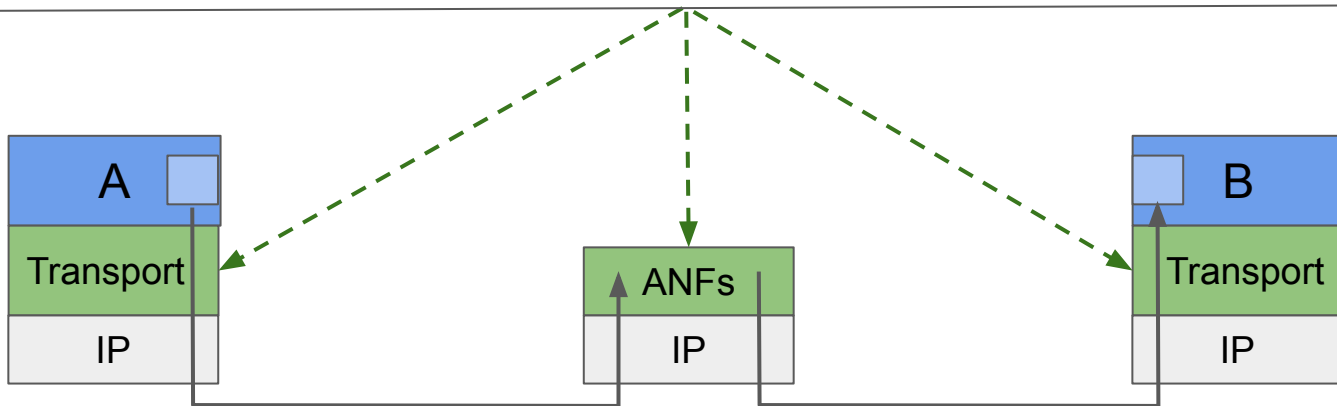
```
GET:  
  name: str  
  user: str
```

Transport [NetBlocks]

```
match rpc-schema:  
  GET => unreliable()  
  SET => reliable()->ordered()
```

ANFs [AppNet]

```
firewall() -> session_tracker()  
  
firewall():  
  user = get(rpc, 'user')  
  match get(firewall_rules, user):  
    ALLOW => send(rpc)  
    DENY|NONE => drop()  
  ...
```



ANF-aware RPC serialization

```
rpc:  
  name: str  
  user: str
```

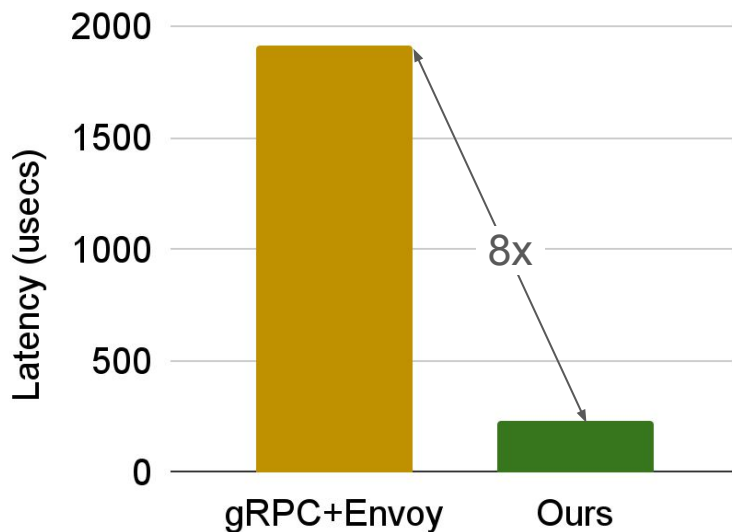
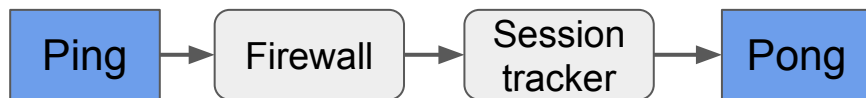
```
firewall():  
  user = get(rpc, 'user')  
  ...
```



user.len	user.val	name.val
----------	----------	----------

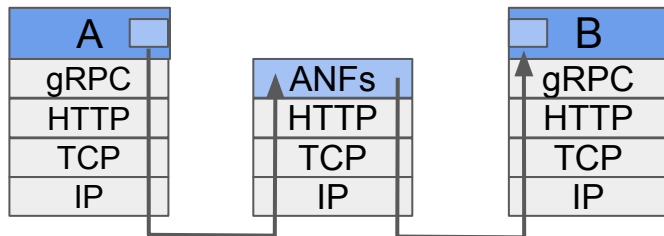
```
firewall:  
  user = rpc[1..rpc[0]]  
  ...
```

(Super) Preliminary results



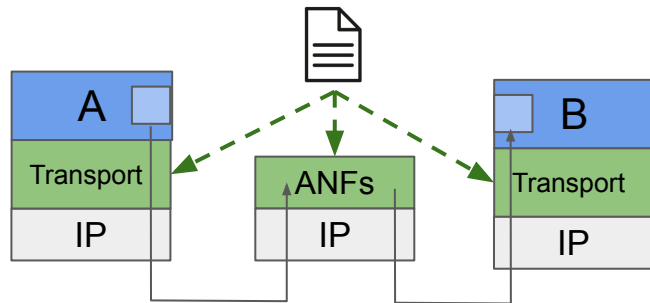
Recap

RPC communication is taxed heavily by protocol layering and loose ANF coupling



We can drive the tax to near zero by

- Specifying requirements at a high level
- Auto-generating a flat stack and ANFs



Building closed-world networks

Alibaba HPN: A Data Center Network for Large Language Model Training

Kun Qian, Yongqing Xi, Jiamin Cao, Jiaqi Gao, Yichi Xu, Yu Guan, Binzhang Fu, Xuemei Shi

RDMA over Ethernet for Distributed AI Training at Meta Scale

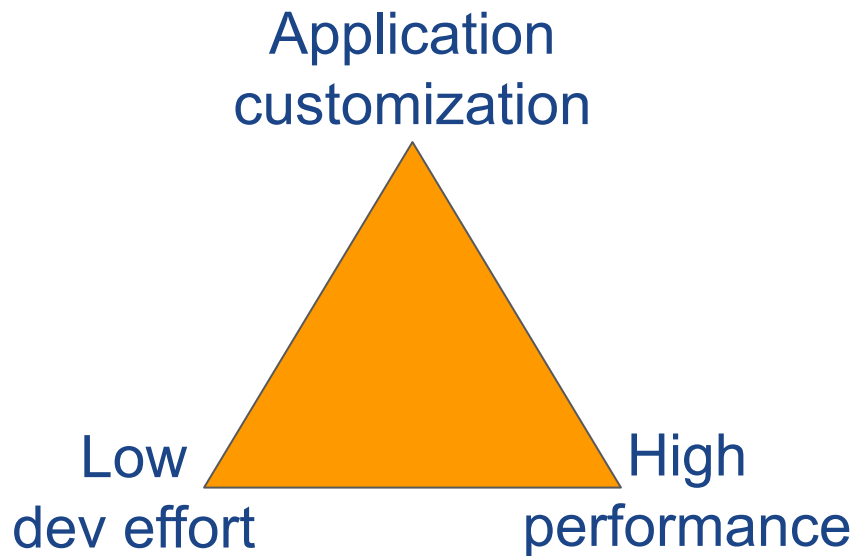
Adithya Gangidi, Rui Miao, Shengbao Zheng, Sai Jayesh Bondu, Guilherme Goes, Hany Morsy, Rohit Puri, Mohit Iyer

ServiceRouter: Hyperscale and Minimal Cost Service Mesh at Meta

Harshit Saokar¹, Soteris Demetriou^{1,2}, Nick Magerko¹, Max Kontorovich¹, Josh Kirstein¹, Margot Leibold¹, Dimitrios Skarlatos^{1,3}, Hitesh Khandelwal¹, and Chunqiang Tang¹

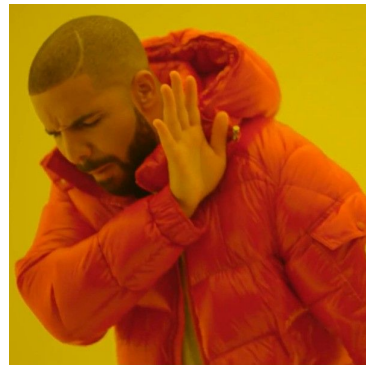
¹ Meta Platforms, ² Imperial College London, ³ Carnegie Mellon University

Building closed-world networks



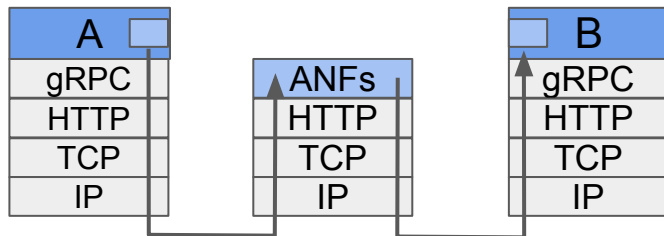
**General
purpose
abstractions**

**Application
defined
networks**



Recap

RPC communication is taxed heavily by protocol layering and loose ANF coupling



We can drive the tax to near zero by

- Specifying requirements at a high level
- Auto-generating a flat stack and ANFs

