A High Performance Packet Core for Next Generation Cellular Networks

Zafar Qazi [+]

Melvin Walls[★], Aurojit Panda[+★], Vyas Sekar[§], Sylvia Ratnasamy[+★], Scott Shenker[+★]

Berkeley UNIVERSITY OF CALIFORNIA [+]

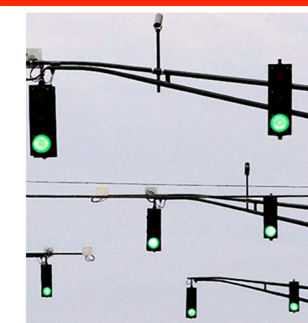
NEFELI [★]

Carnegie Mellon [§]

Explosive Cellular Growth

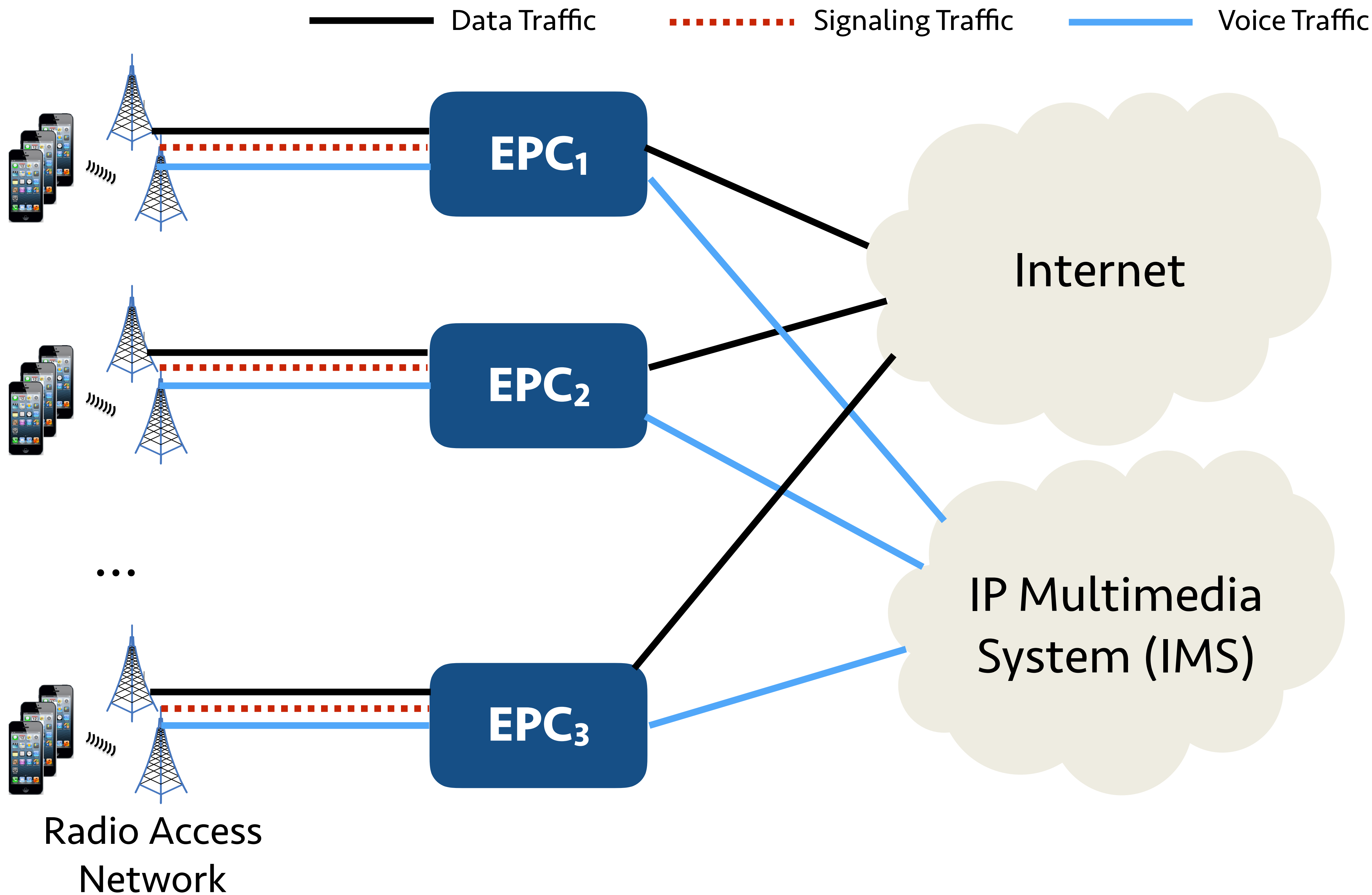
Many Diverse Devices:
3B IoTs by 2019*

Signaling traffic growth:
50% faster growth than data+



Demanding applications
3/4 of data traffic will be video*

Evolved Packet Core (EPC)



Existing Cellular Core Cannot Keep Up!

Concerns from Operators

Operators Urge Action Against Chatty Apps

LONDON -- Open Mobile Summit -- Representatives from some of Europe's largest operators called for action here on Wednesday to prevent mobile applications from overloading their networks with signaling traffic.

Operators Fight Back on Smartphone Signaling

Mobile operators have demanded that smartphone makers support a newly standardized feature in their devices that will reduce the signaling traffic load on their networks.

Academic Studies

Understanding the bottlenecks in Virtualizing Cellular Core Network Functions

Ashok Sunder Rajan[*], Sameh Gobriel[*], Christian Maciocco[*], Kannan Babu Ramia[*], Sachin Kapur[†], Ajaypal Singh[†], Jeffrey Erman[‡], Vijay Gopalakrishnan[‡] and Rittwik Jana[‡]
[*]Intel Labs, [†]Connectem, [‡]AT&T Labs

Considerations for Re-Designing the Cellular Infrastructure Exploiting Software-Based Networks

(Invited Extended Abstract)

Ali Mohammadkhan[*], K. K. Ramakrishnan[*], Ashok Sunder Rajan[‡], and Christian Maciocco[‡]
[*]University of California, Riverside [‡]Intel Labs

Industrial Efforts

CORD®

Central Office Re-architected as a Datacenter

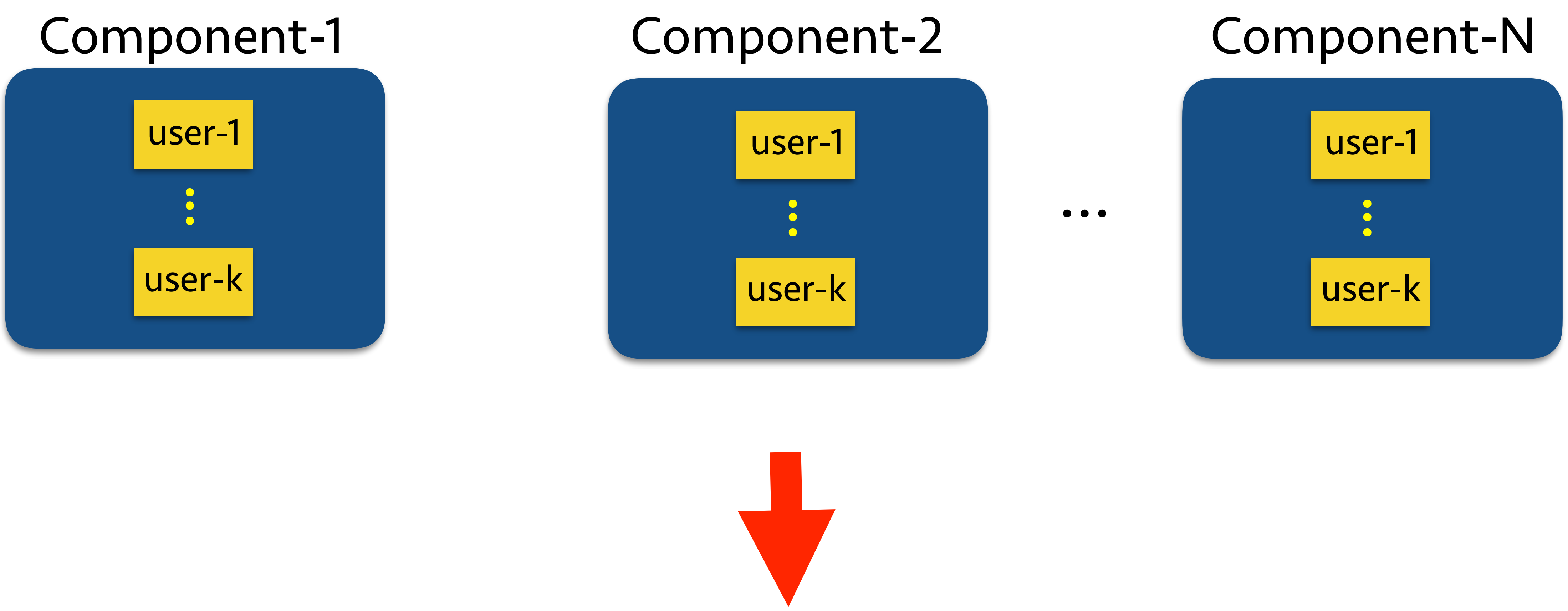
Sprint, Intel Join Forces on C3PO 5G User Plane Open Source Project

Core Blimey! Affirmed Eyes 5G Services Evolution Next

Virtualization startup Affirmed is getting ready to show off its take on 5G core network software, which will involve separating user control functions from a centralized control plane.

The Cisco 5G Strategy Series: Packet Core, Transport, and Identity Management

EPCs are factored based on functions



Distributed user state

Performant **EPC** (**PEPC**)

PEPC: EPC functions factored around *state*

An *abstraction* of independent and customizable *EPC slices*

Rest of the Talk …

- Scalability challenges
- Design of PEPC
- Implementation and Evaluation

Traditional EPCs

EPC

Signaling Function

- Mobility Management Entity (MME)

Data Gateways

- Serving Gateway (S-GW)
- Packet Data Network Gateway (P-GW)

Backend servers

Subscriber Database

- Home Subscriber Server (HSS)

Policy Server

- Policy and Charging Rules Function (PCRF)

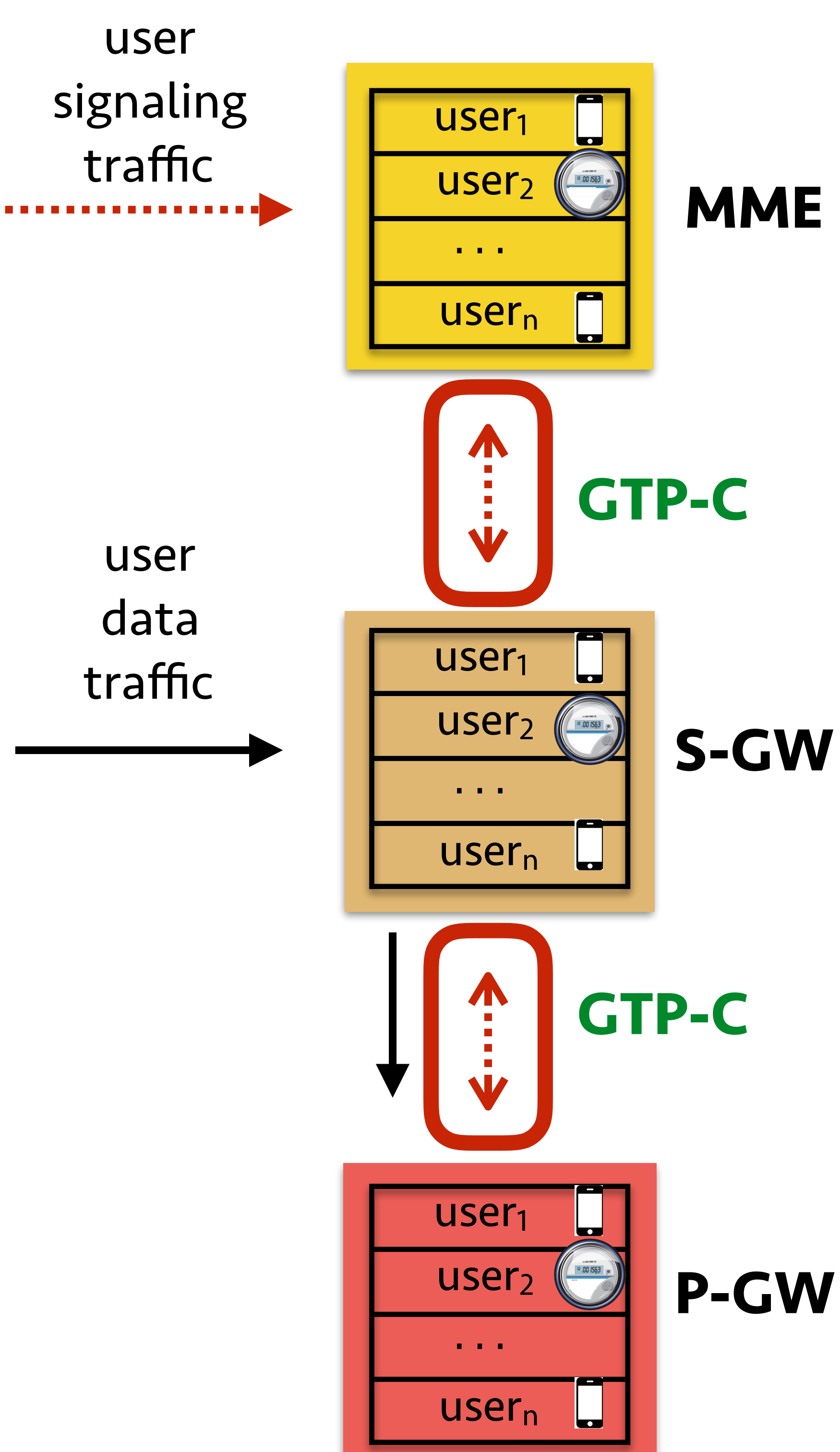
Implemented as hardware appliances

Statically provisioned at a few central locations

User state in EPC

State Type	MME	S-GW	P-GW	Update Frequency
Per-user QoS/policy state	w+r	w+r	w+r	per-signaling_event
User id	w+r	w+r	w+r	per-signaling_event
User location	w+r	w+r	NA	per-signaling_event
Per-user control tunnel state	w+r	w+r	w+r	per-signaling_event
Per-user data tunnel state	w+r	w+r	w+r	per-signaling_event
Per-user bandwidth counters	NA	w+r	w+r	per-packet

Distributed User State is Problematic!

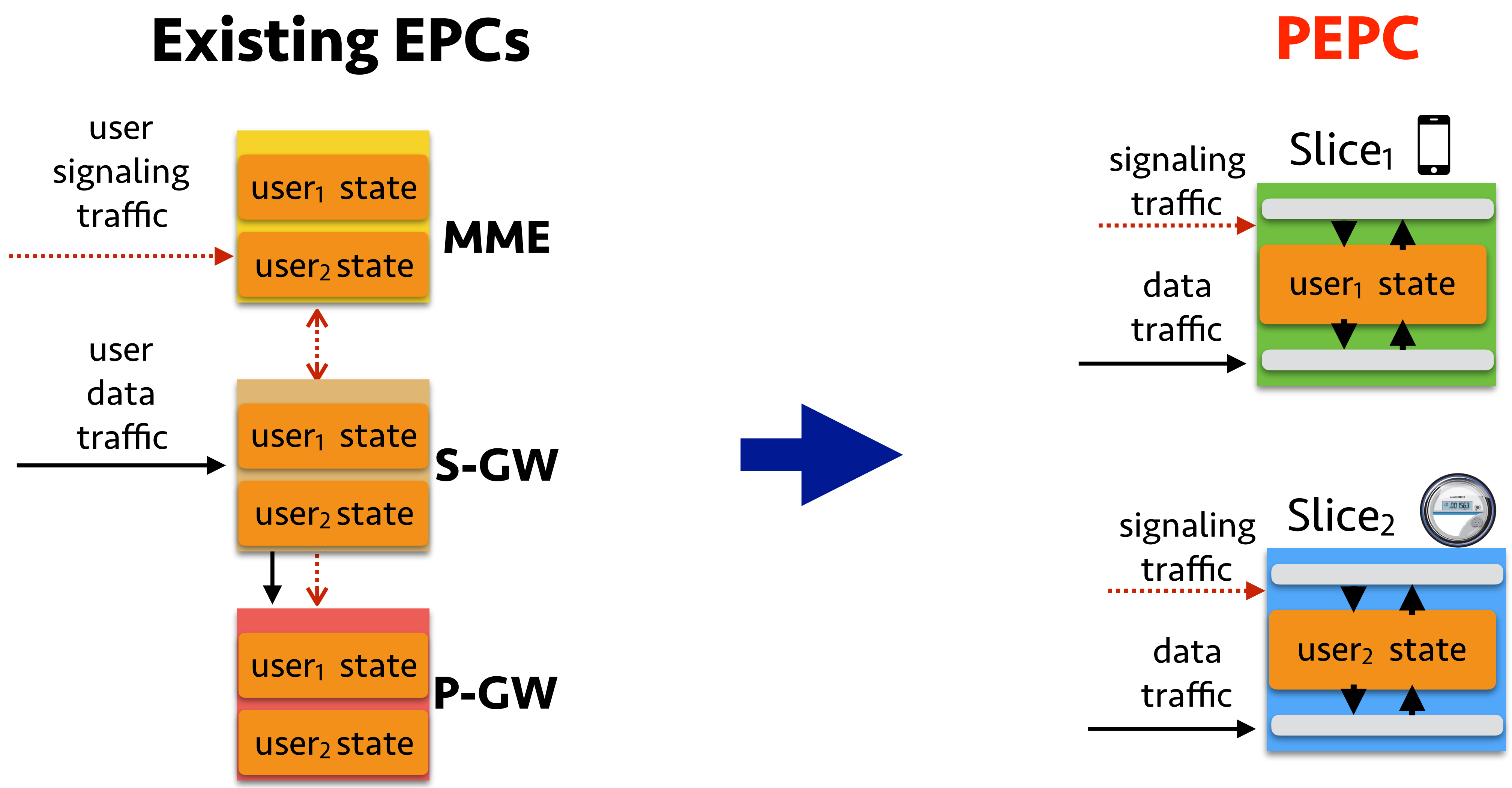


- Performance overheads + high complexity
  - Frequent cross component synchronization
- Migration is hard
  - Distributed user state
- Customization is hard
  - Distributed user state + distributed computation

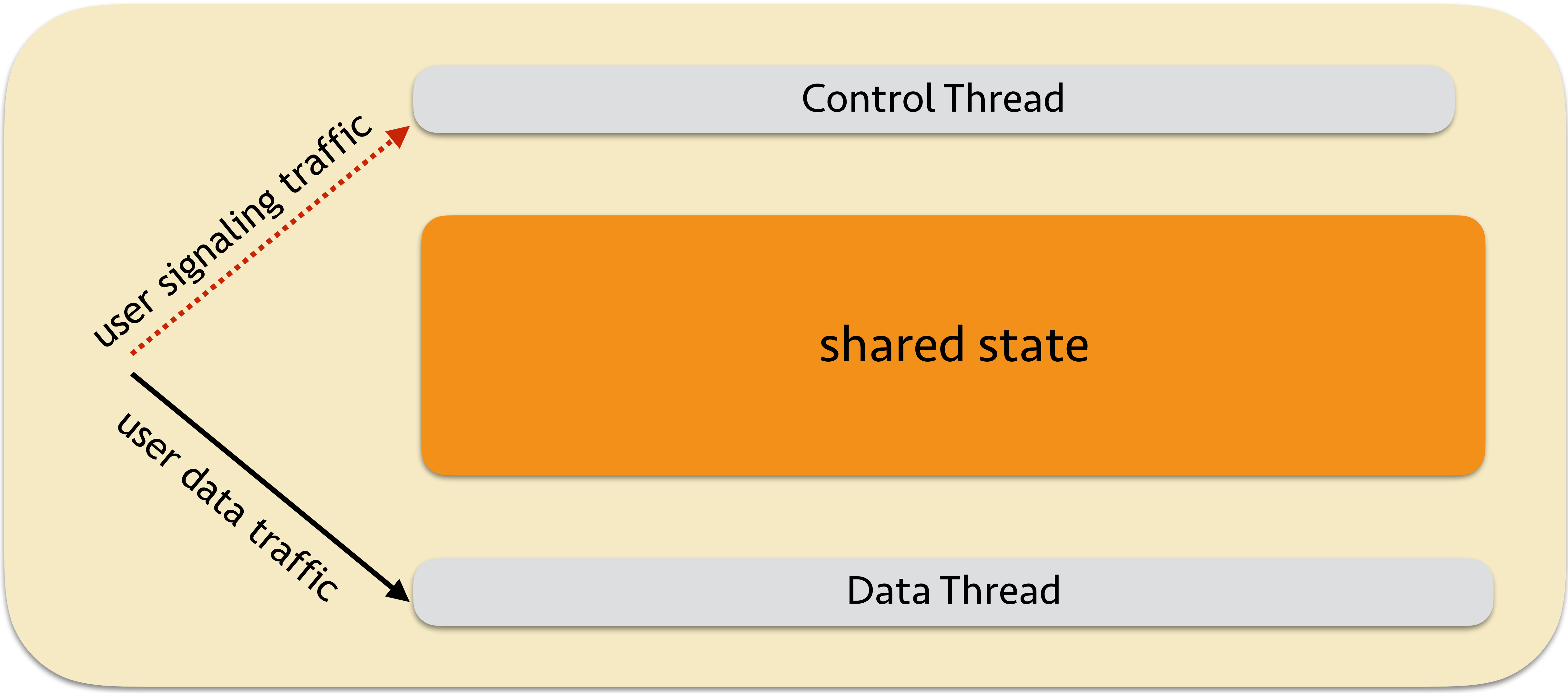
Rest of the Talk …

- Scalability challenges
- Design of PEPC
- Implementation and Evaluation

Existing EPC vs. PEPC



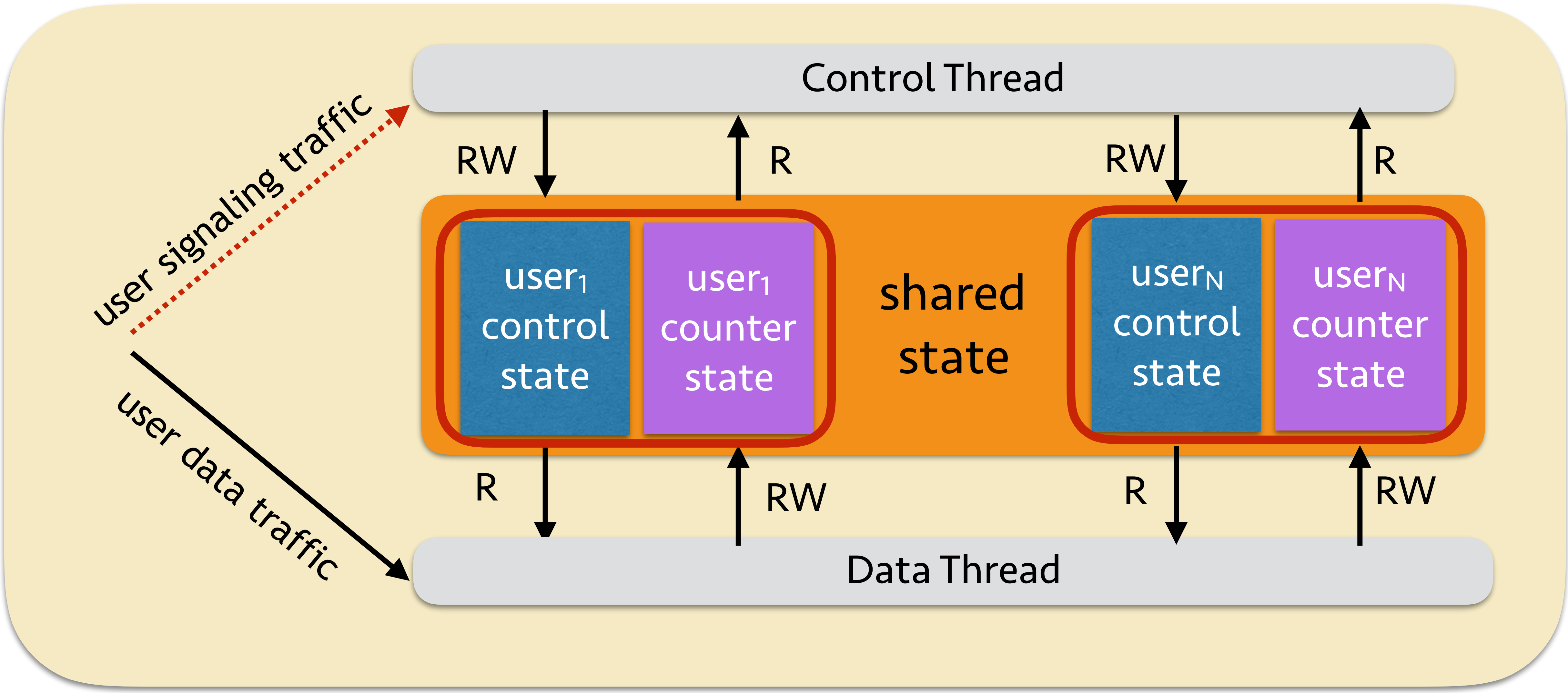
PEPC Slice



Separation of control and data thread —> *avoid HOL blocking*

- Processing time for signaling messages > 10X higher
- Control and data threads assigned to separate cores

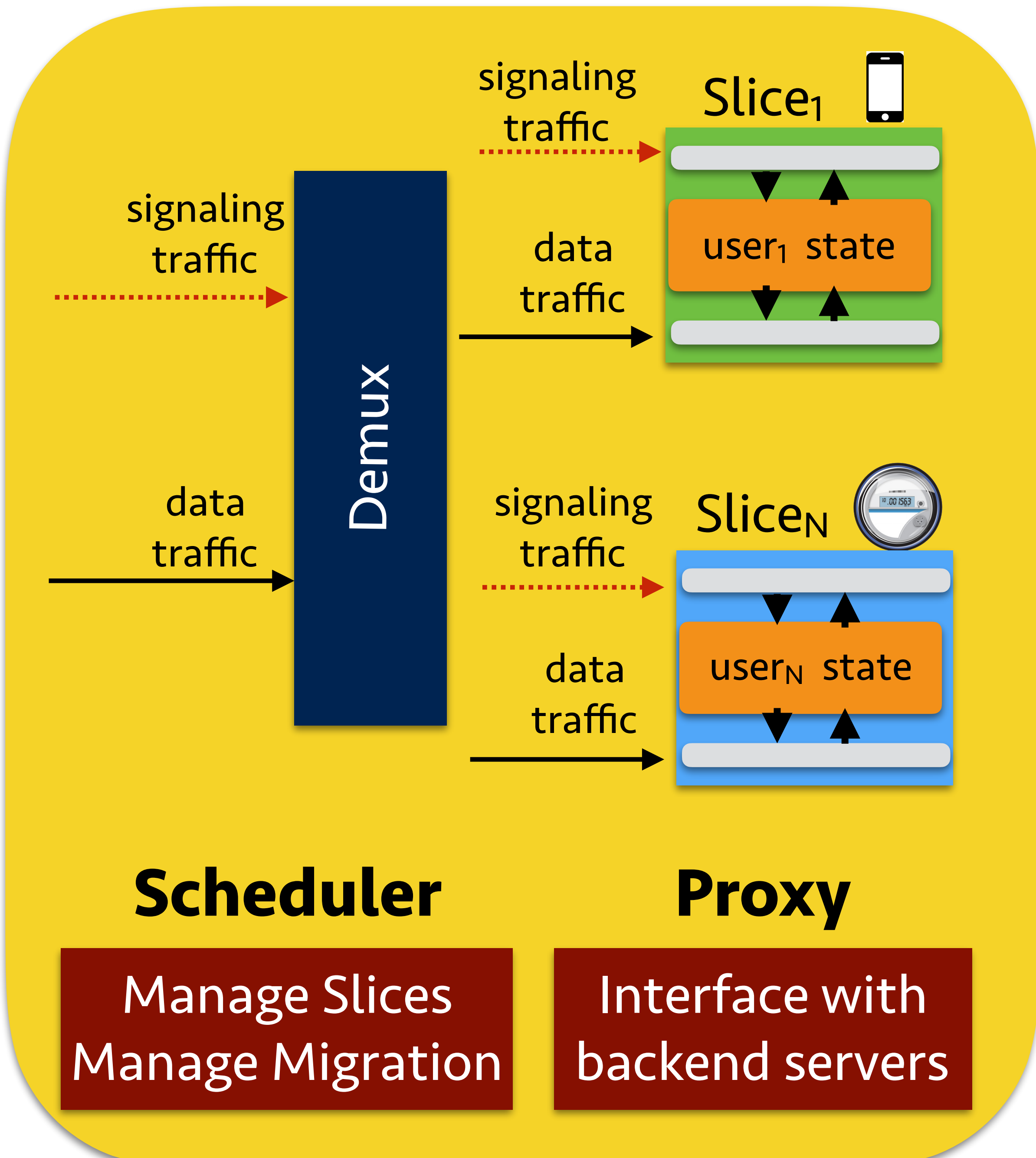
PEPC Slice



Partition shared state at two levels —> *reduces contention*

- By user
- Per-user state whether control or data thread writes to it
- Use fine grained locks —> *up to 5X improvement over coarse grained locks*

PEPC Server



- Pause + snapshot user state
  —> simplifies state migration
- Modify slice data/control flow
  —> simplifies customization

Rest of the Talk …

- Scalability challenges
- Design of PEPC
- Implementation and Evaluation

Implementation

- Data plane functions
  - GPRS Tunnelling Protocol (GTP)
  - Policy and charging enforcement function
- Signaling functions
  - Implements, S1AP, the protocol used to interface with the base stations
  - Supports state updates for attach request
- Support for efficient state migration across slices
- PEPC uses the NetBricks* programming framework

** Panda et al. NetBricks: Taking the V out of NFV. OSDI'16*

PEPC Customization/Optimization Examples

- **Two level user state storage**

Active devices

Attached but ***Inactive*** devices

Improves state lookup performance for data packets

- **Customization** for a class of IoT devices (like smart meters)
  - Devices that run a single application
  - Reduce state and customize data processing

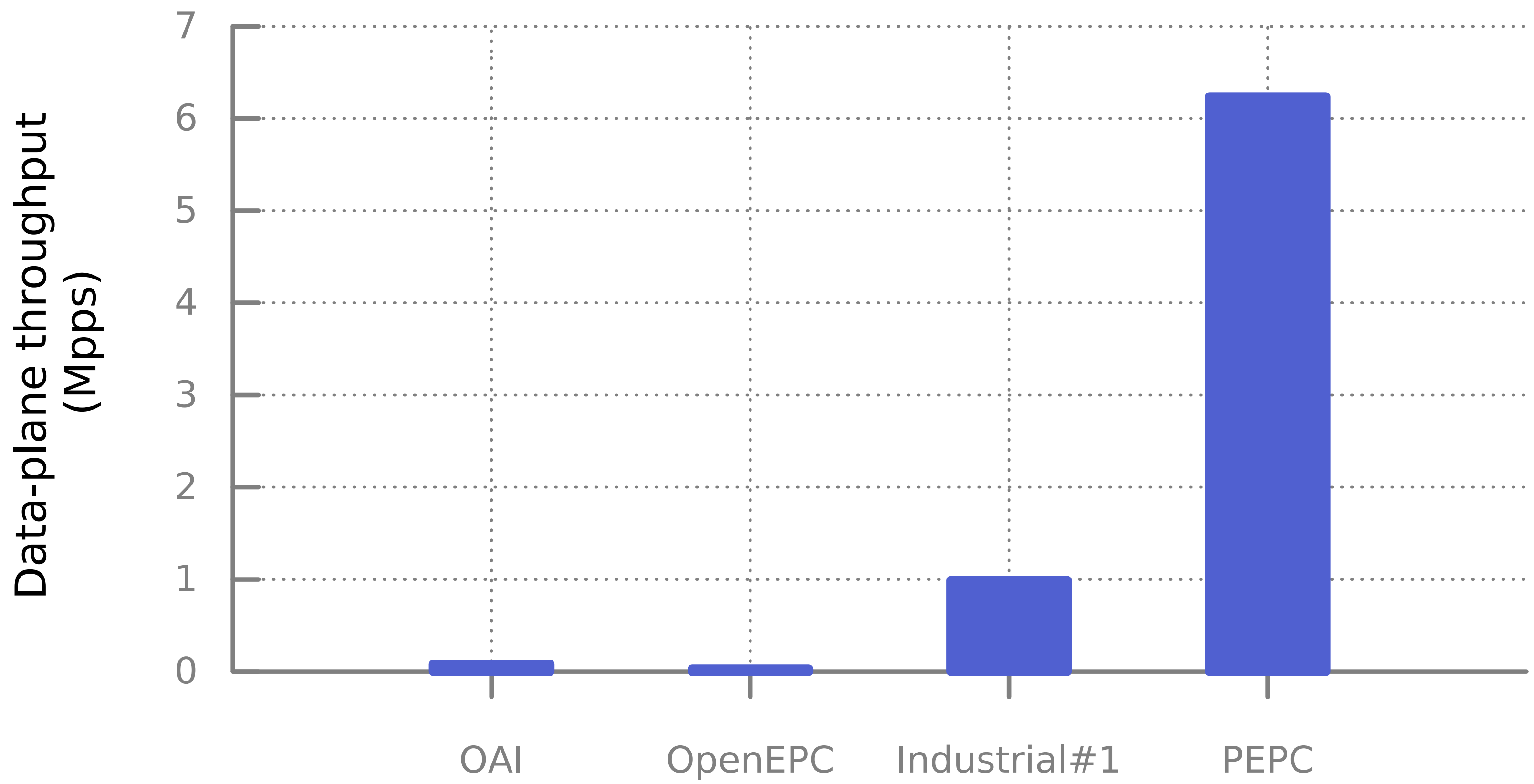
Evaluation and Methodology

- How does PEPC compare with other vEPCs?
- How scalable is PEPC with increasing signaling traffic?
- How scalable is PEPC is with increasing state migrations?
- Benefits of PEPC's customization/optimizations?
- Methodology
  - DPDK based traffic generator
  - Replays cellular data and signaling traffic traces
  - Traces from OpenAirInterface and ng4T

Baselines

- Industrial#1: An industrial software EPC implementation
  - DPDK based
  - Ran as a process inside the host OS, not inside a VM/Container
  - S-GW/P-GW on one server, and MME on an another server
- OpenAirInterface (OAI): An opensource EPC software
  - Ran as a process inside the host OS
  - Ran all EPC functions on the same server
- OpenEPC on Phantomnet: Software EPC implementation
  - MME, S-GW, P-GW on different servers

Data plane performance comparison



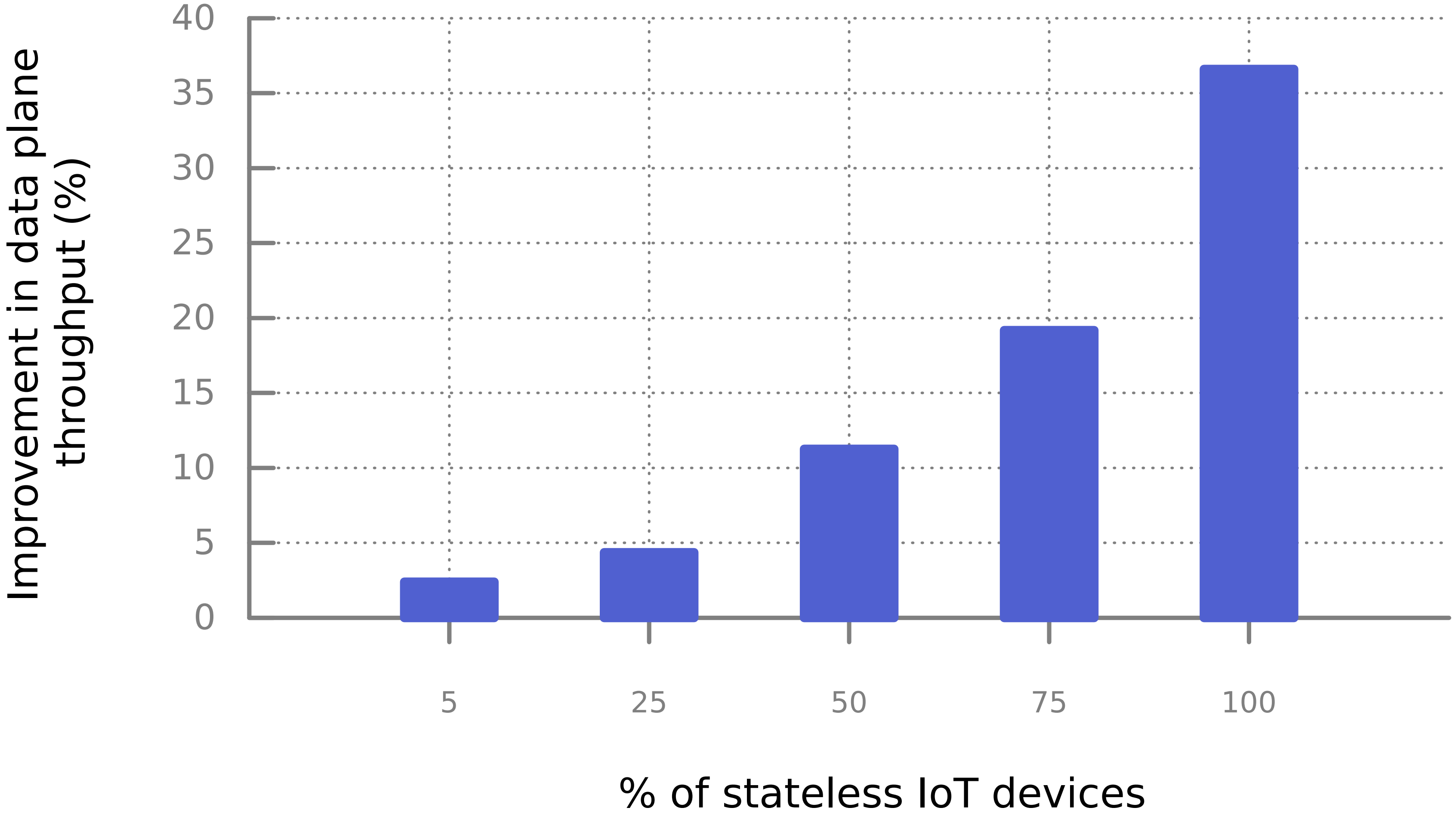
250K users

10K attach-requests/s

PEPC can sustain data plane throughput (~6 Mpps) for 1:10 signaling to data ratio

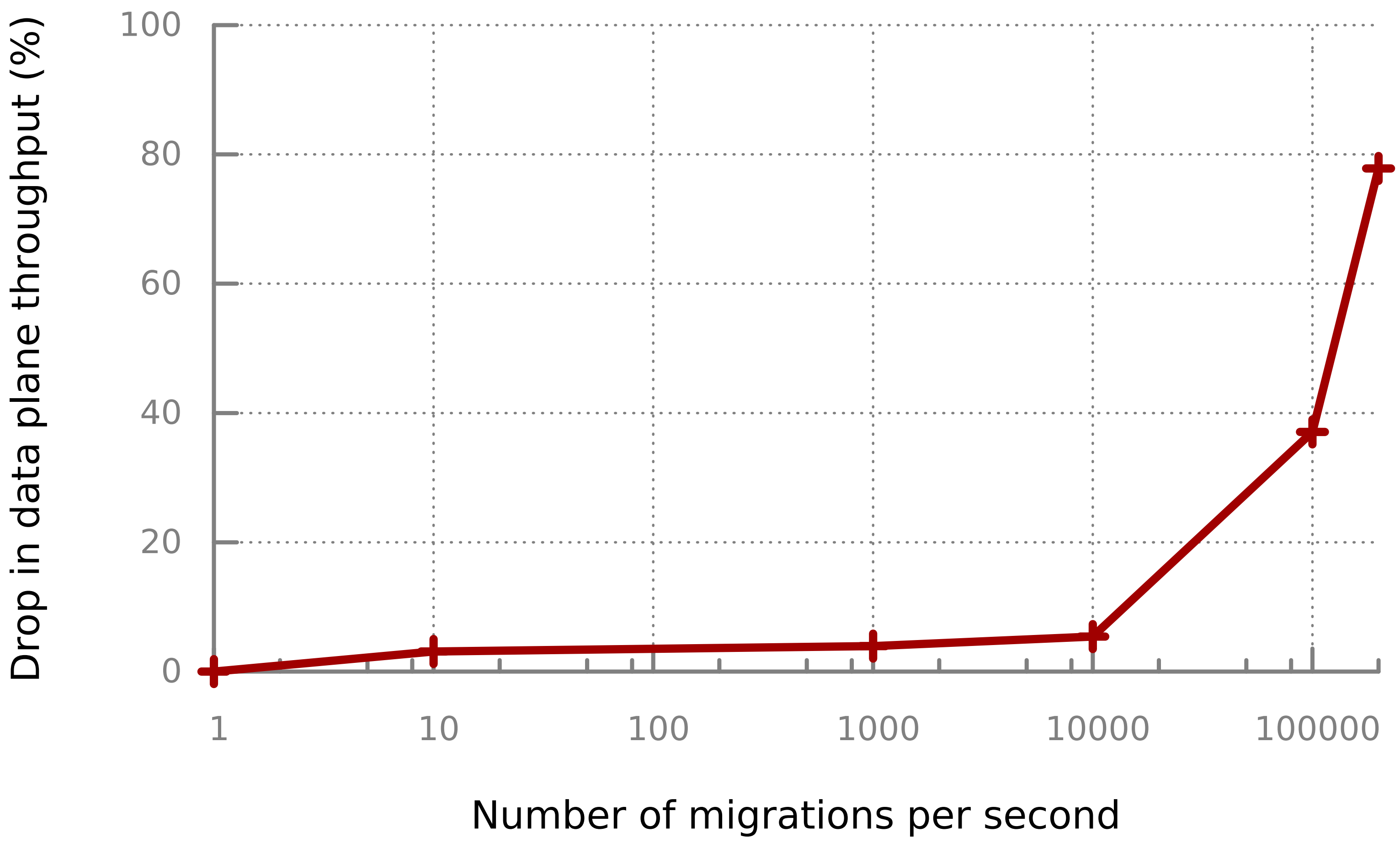
In contrast, *Industrial#1* throughput drops significantly (0.1 Mpps) for more than 1:100 signaling to data ratio

PEPC Customization Benefits



For smart meter like devices, can achieve up to 38% improvement

Scalability with State Migrations



- State migration across two slices within a server

Less than 5% drop in data plane throughput with 10K migrations per sec

Related work

- SDN based cellular designs
  - SoftCell [*CoNEXT'13*], SoftMoW [*CoNEXT'15*]
- Virtual EPCs
  - KLEIN [*SOSR'16*], SCALE [*CoNEXT'15*]

Summary

- Existing EPC systems cannot keep up with cellular growth
  - Key reason: user state is distributed
- New system architecture: PEPC
  - Refactoring of EPC functions based on user state
  - Enables horizontal slicing of EPC by users into independent and customizable slices
- PEPC performs 3-7x better and scales well with increasing user devices, signaling traffic, and state migrations