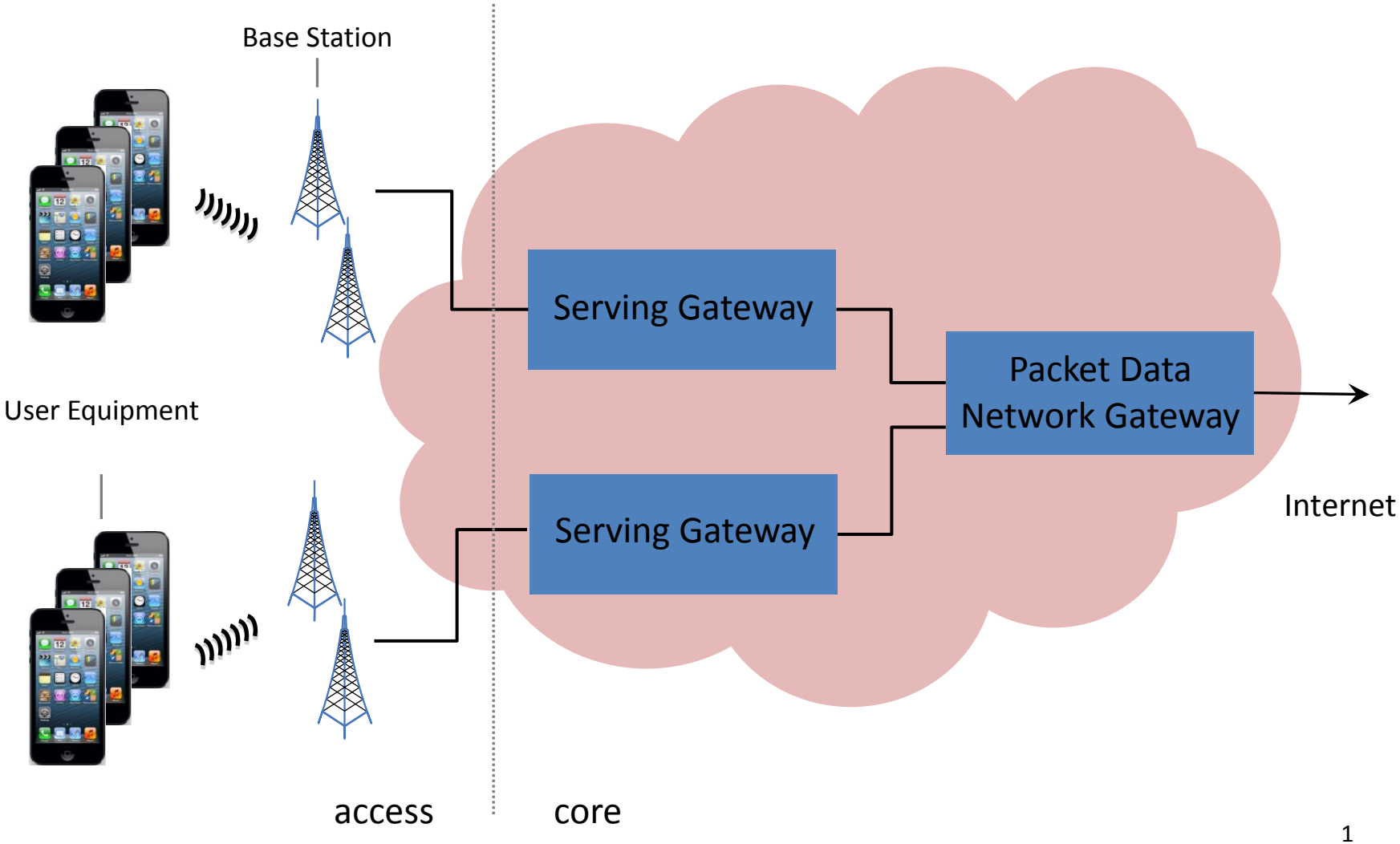
CellSDN: Software-Defined Cellular Core networks

Xin Jin

Princeton University

Joint work with Li Erran Li, Laurent Vanbever, and Jennifer Rexford

Cellular Core Network Architecture

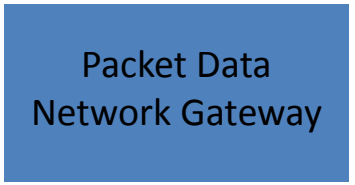


Cellular core networks are not flexible

- Most functionalities are implemented at

  Packet Data Network Gateway

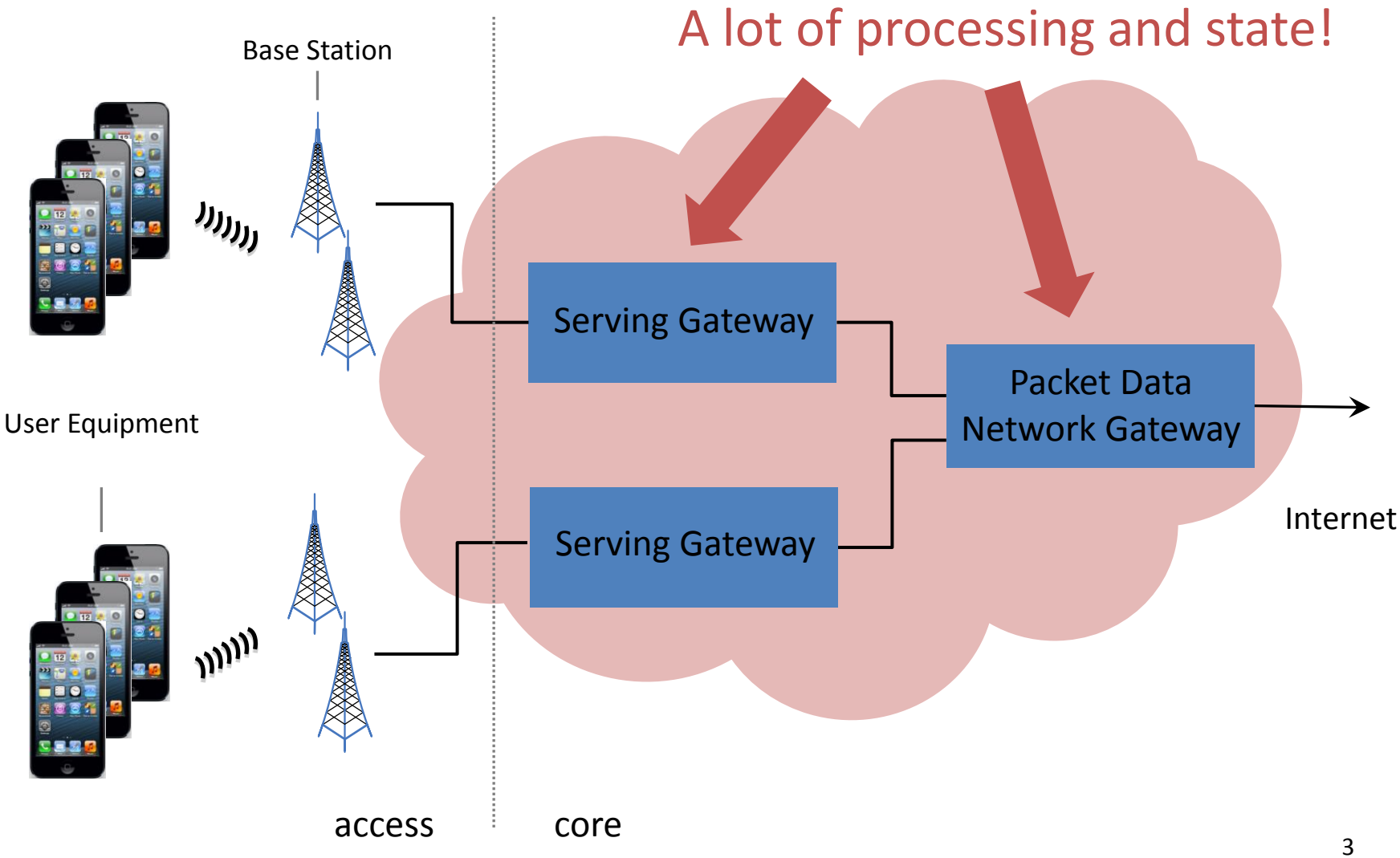
  – Application identification, content filtering (DPI), monitoring and billing, …



- This is not flexible

Combine functionalities from different vendors	✖
Easy to add new functionality	✖
Only expand capacity for bottlenecked functionality	✖

Cellular core networks are not scalable



Cellular core networks are not cost-effective

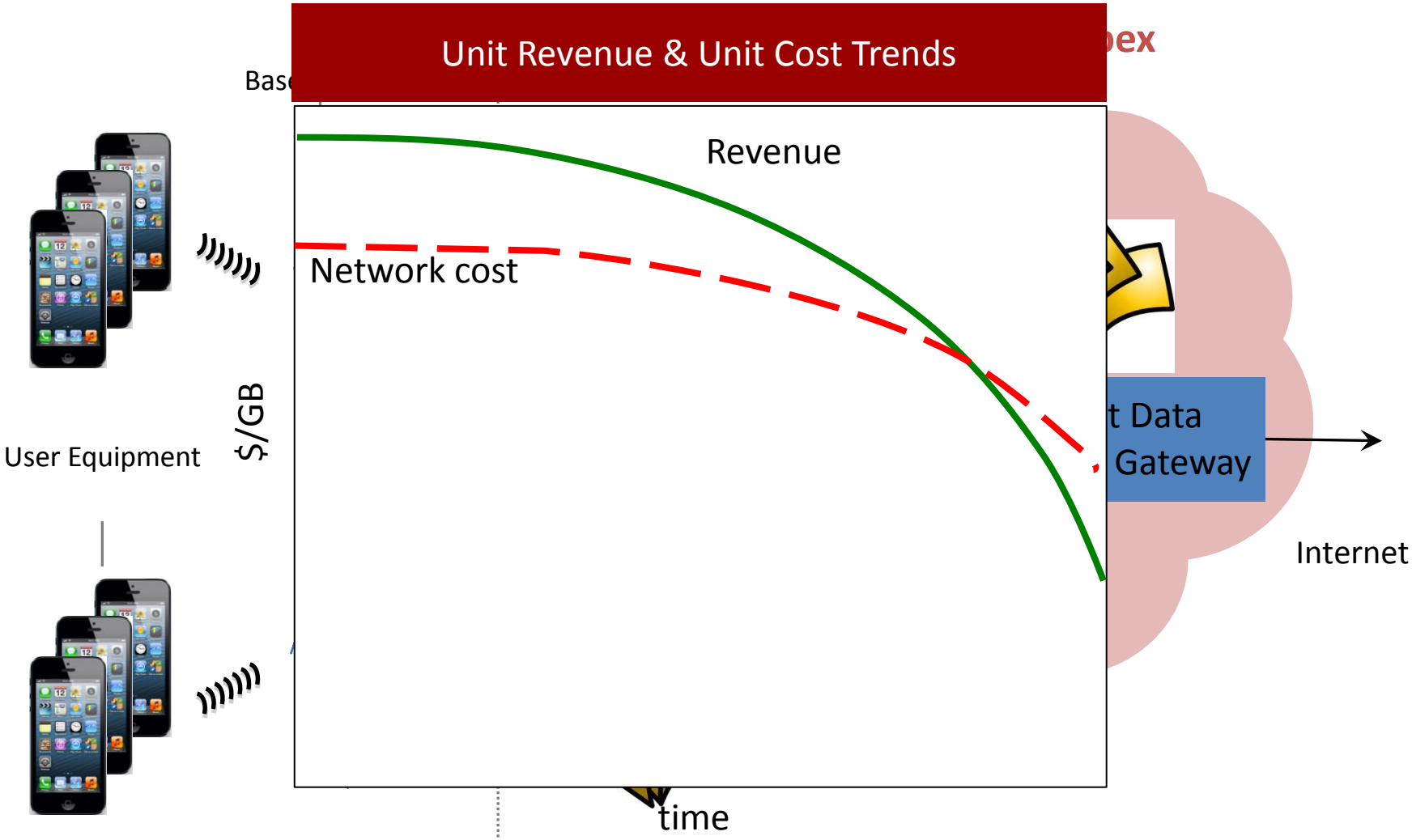


Figure from S. Elby, "Carrier vision of SDN and future applications to achieve a more agile mobile business,", Keynote at the SDN & OpenFlow World congress, October 2012.

Can we make cellular core networks like data center networks?

- ✔ *Flexible*
- ✔ *Scalable*
- ✔ *Cost-Effective*

Can we make cellular core networks like data center networks?

Yes! With CellSDN!

✔ *Flexible*
✔ *Scalable*
✔ *Cost-Effective*

Characteristics of Cellular Core Networks

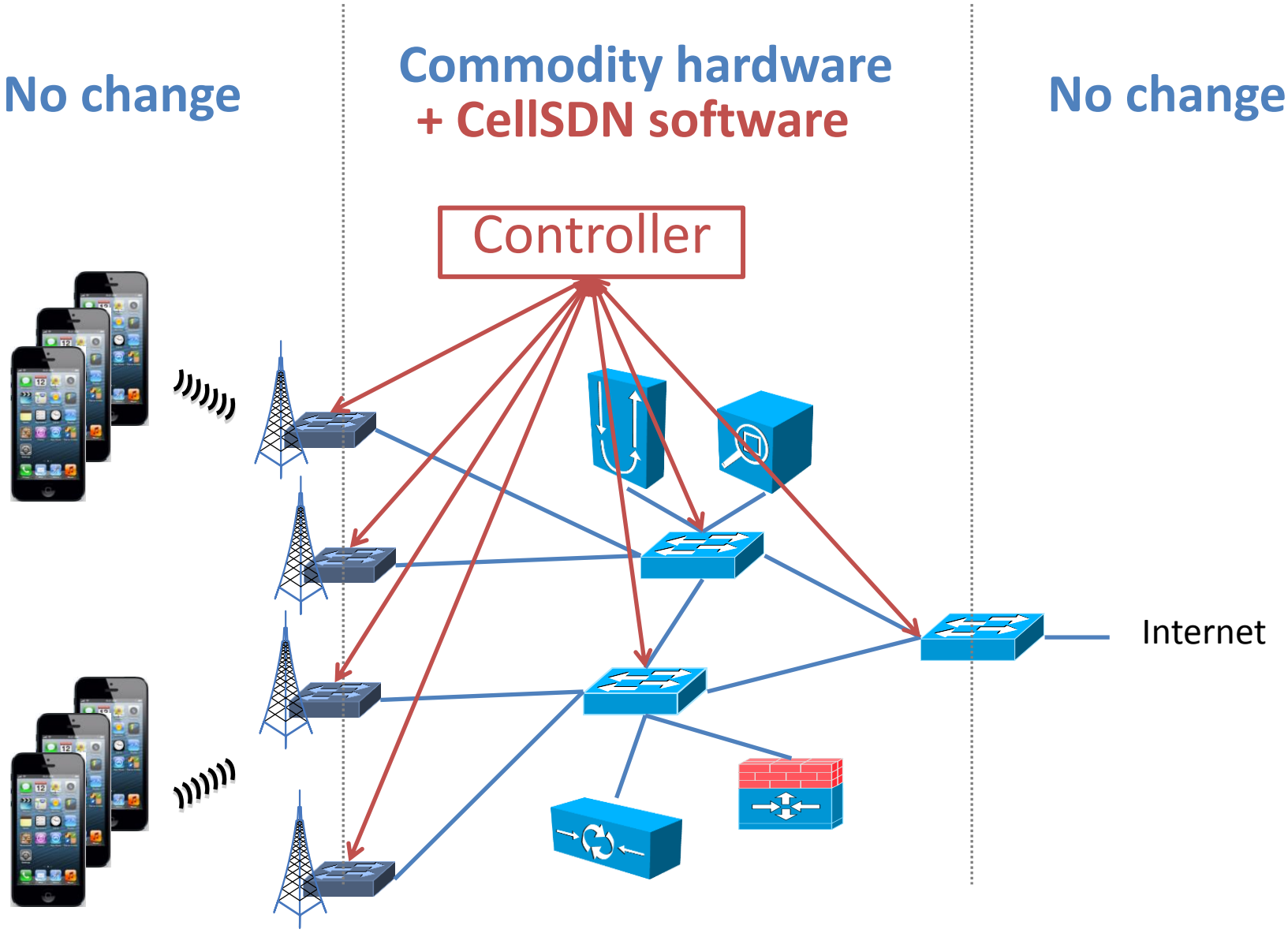
1. Fine-grained and sophisticated policies

with diverse needs!

2. "North south" traffic pattern: in cellular core networks, most traffic is from/to the Internet
   - In data centers, 76% traffic is intra data center traffic. [Cisco Global Cloud Index]

3. Asymmetric edge: low-bandwidth access edge vs. high-bandwidth gateway edge

CellSDN Overview



Fine-grained and sophisticated policies

" I want

video traffic to gold plan customer to
go through a firewall then a video transcoder

and

balance the load among
all transcoders and firewalls in the network!"

Decouple the problem

" I want

video traffic to gold plan customer to go through a firewall than a video transcoder

and

balance the load among all transcoders and firewalls in the network!"

Decouple the problem

Service Policy: meet customer demand

application type + subscriber attributes
➔ an ordered list of middleboxes

balance the load among all transcoders and firewalls in the network

Decouple the problem

Service Policy: meet customer demand

application type + subscriber attributes
➔ an ordered list of middleboxes

Traffic Management Policy: meet operational goal

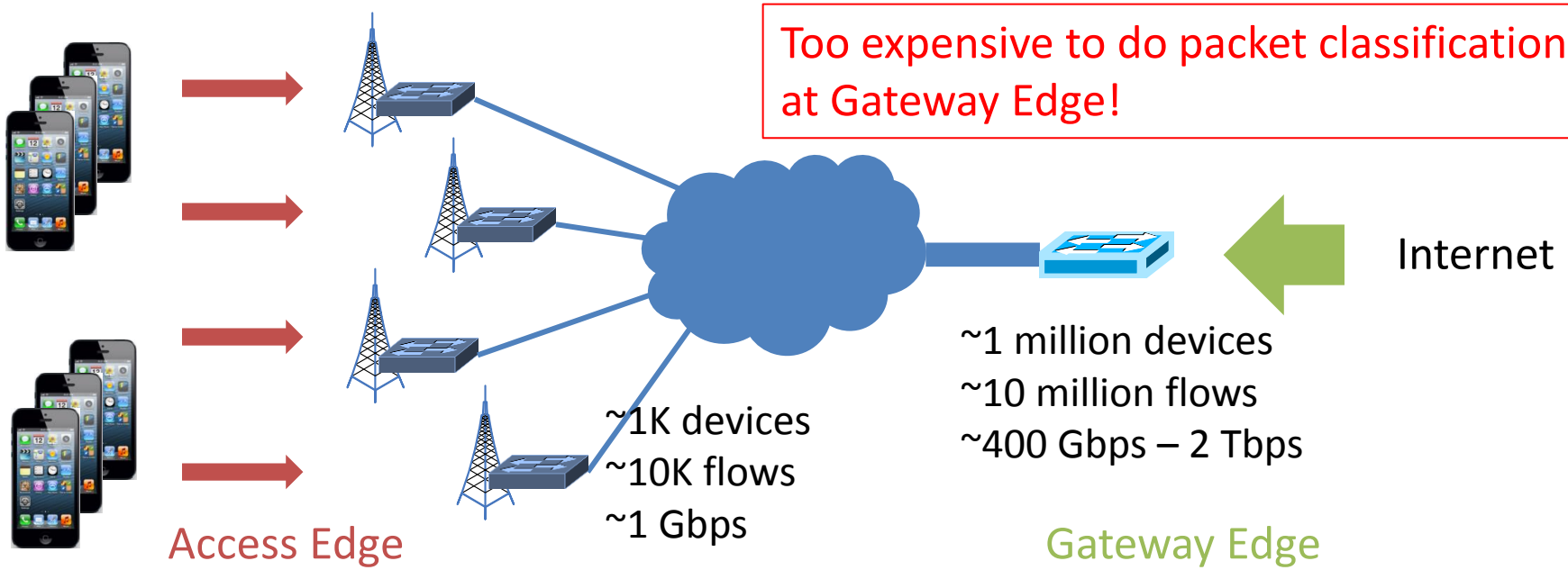
network resource allocation: e.g. load balance among multiple middlebox instances

Challenge: Scalability

- Packet Classification: decide which service policy to be applied to a flow
  - How to classify millions of flows?

- Path Implementation: generate switch rules to implement paths given by traffic management policy
  - How to implement millions of paths?

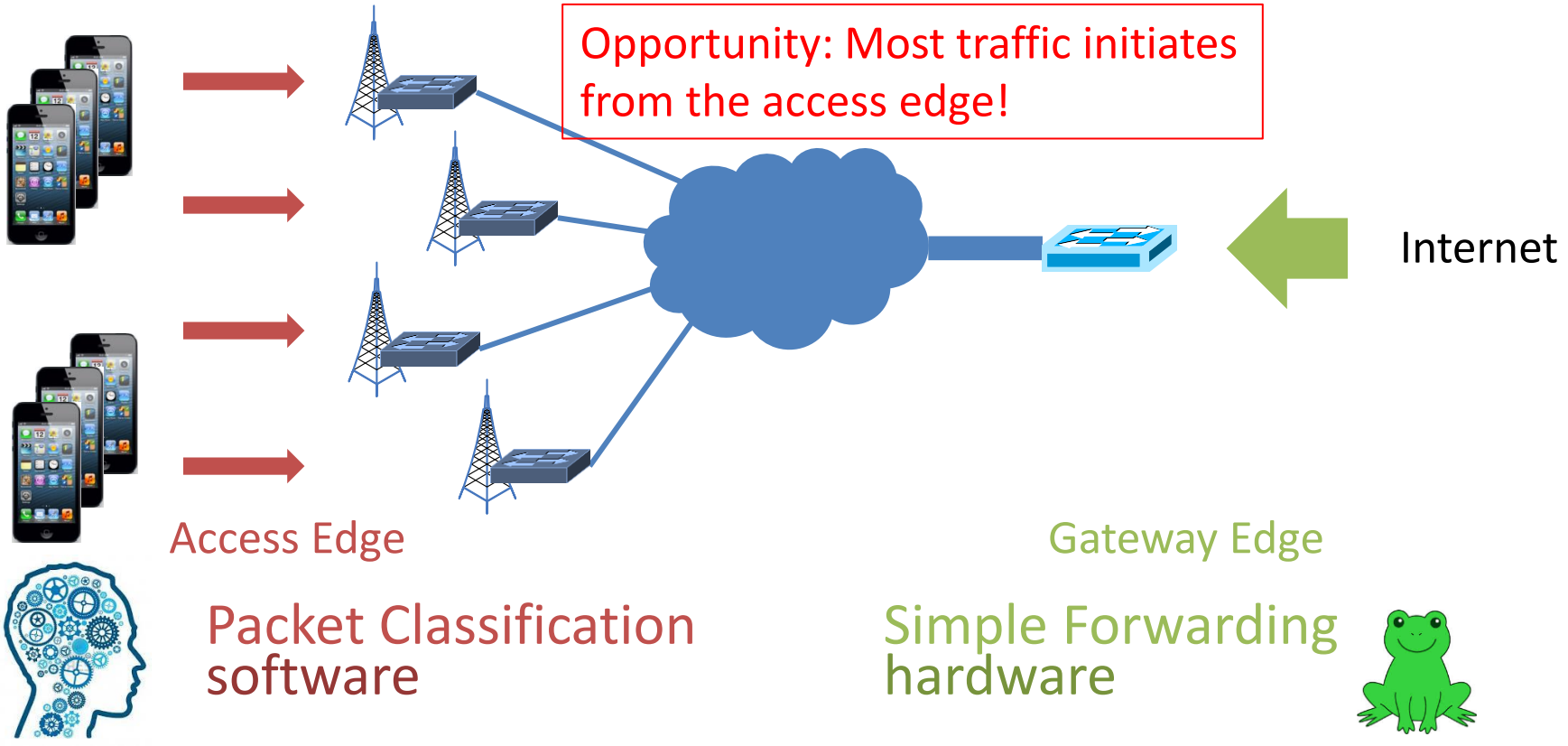
“North south” Traffic Pattern



- Low link speed
- Small number of active flows

- High link speed
- Huge number of active flows

Asymmetric Edge: Packet Classification



- Encode classification results in packet header

- Classification results are implicitly piggybacked in header

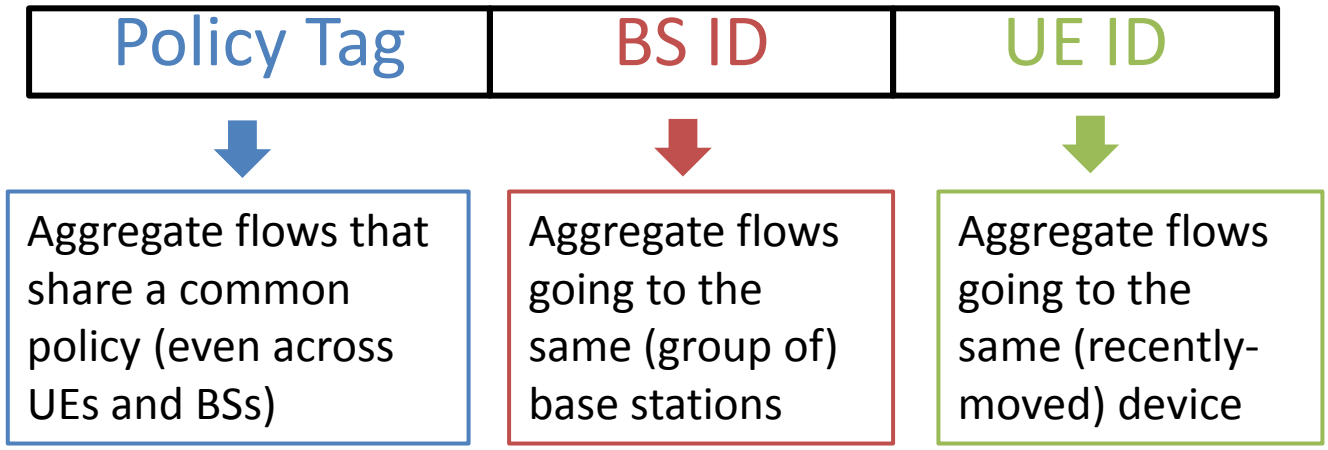
Challenge: Scalability

- Packet Classification: decide which service policy to be applied to a flow
  - How to classify millions of flows?

- Path Implementation: generate switch rules to implement paths given by traffic management policy
  - How to implement millions of paths?

Multi-Dimensional Aggregation

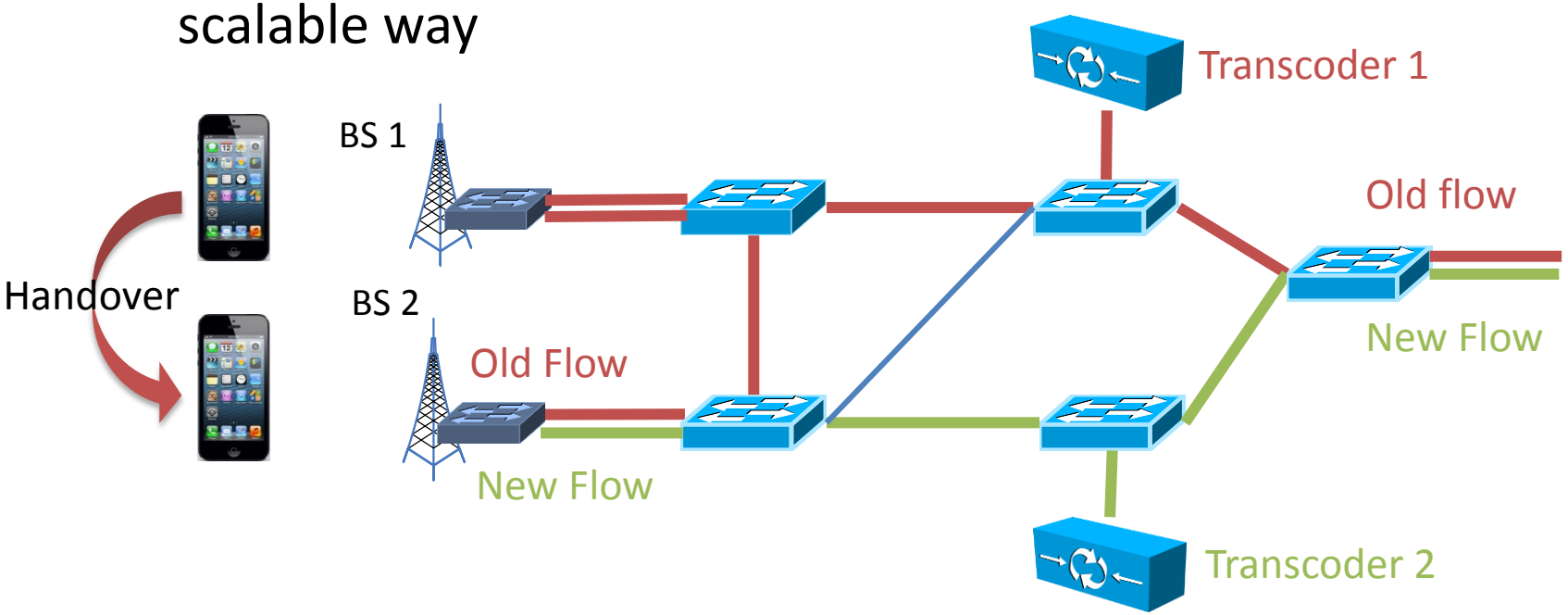
- Use multi-dimensional tags rather than flat tags



- Selectively match on one or multiple dimensions
  - Supported by TCAM in today's switches

Policy Consistency Under UE Mobility

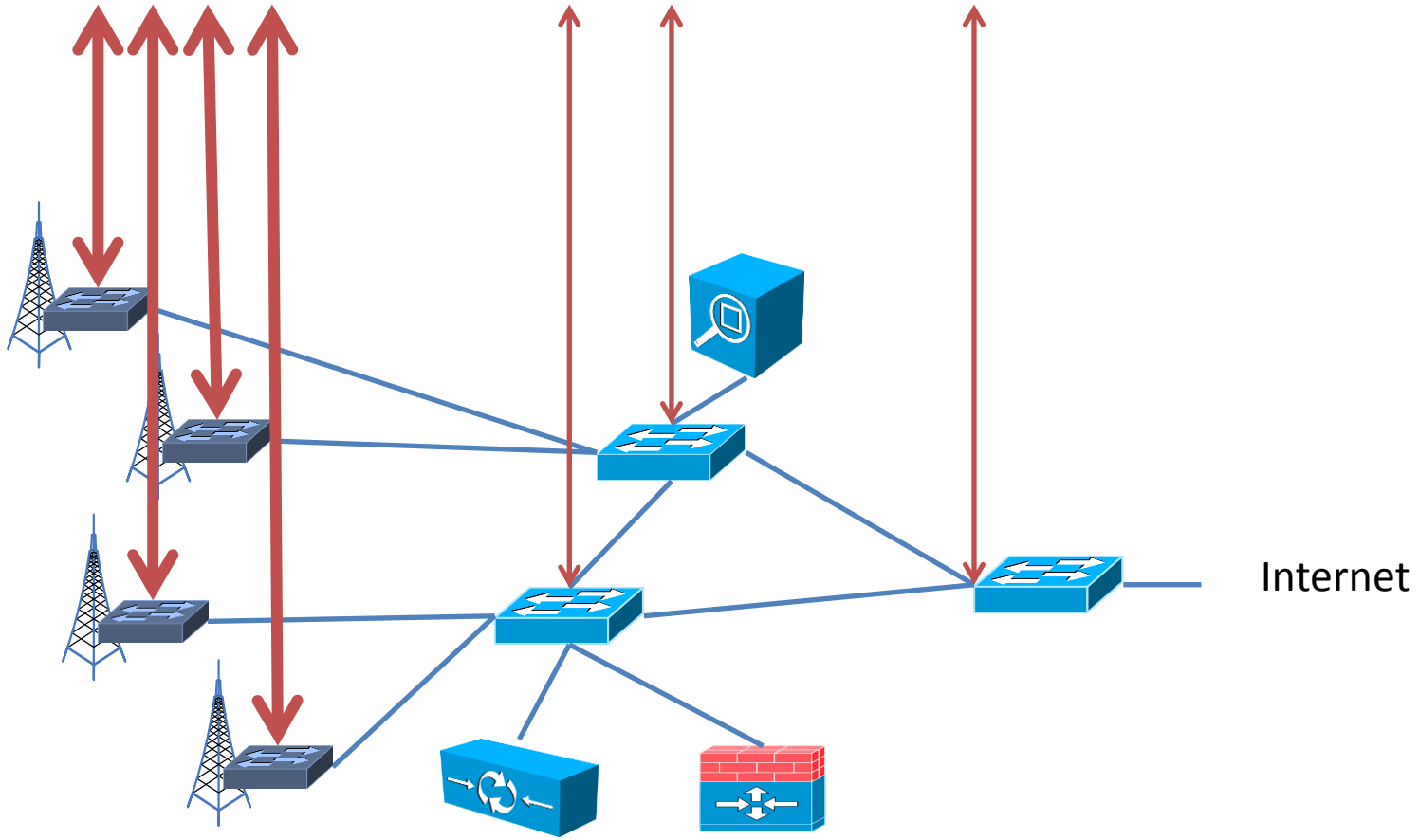
- UE Mobility: frequent and unplanned
- Policy consistency: crucial for stateful middleboxes
  - All packets of a flow go through the same middlebox instances even in the presence of mobility
  - Multi-dimensional tags ensures policy consistency in a scalable way



Control Plane Load

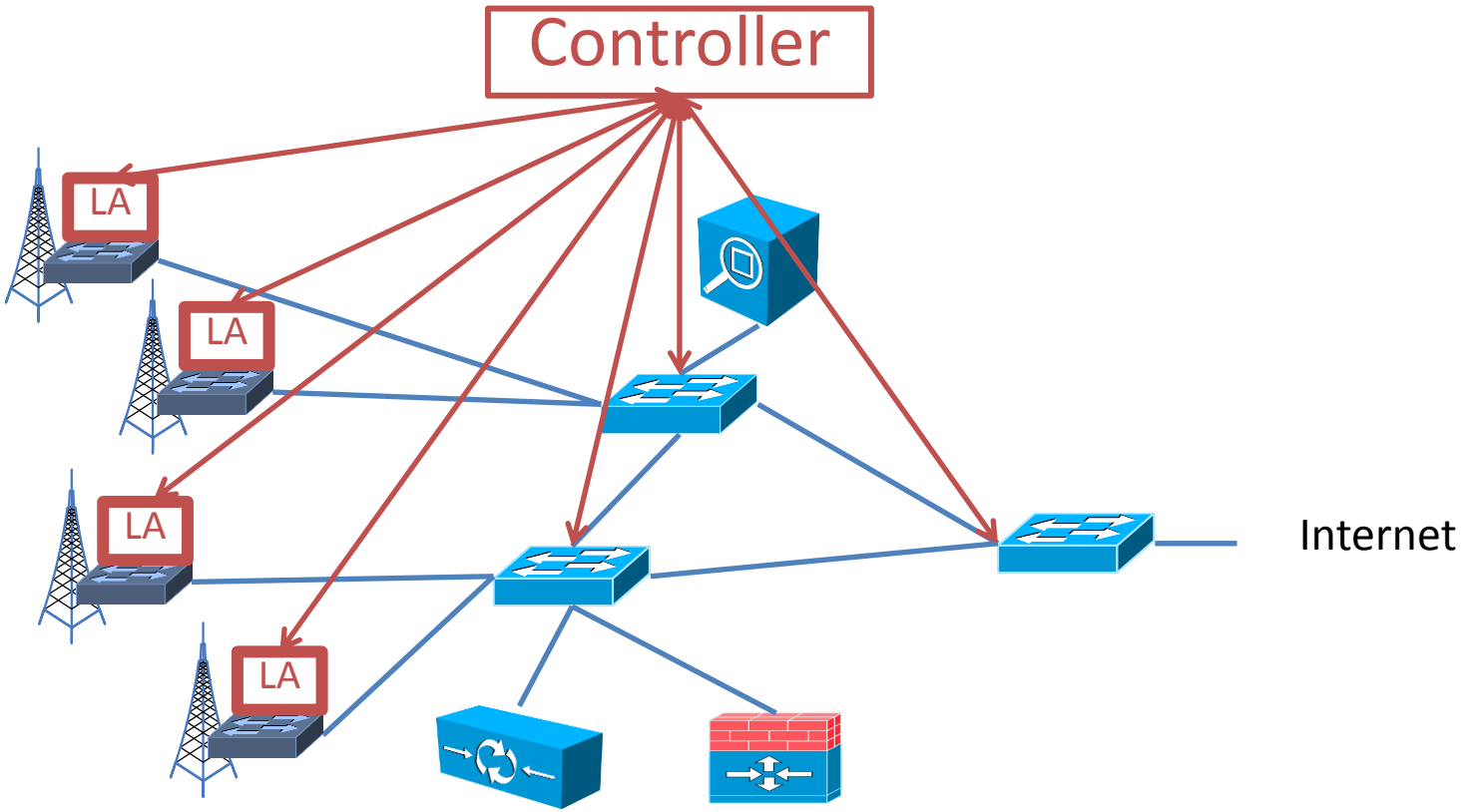
Packet classification
Handle every flow
Frequent switch update

Policy path setup
Handle every policy path
Infrequent switch update



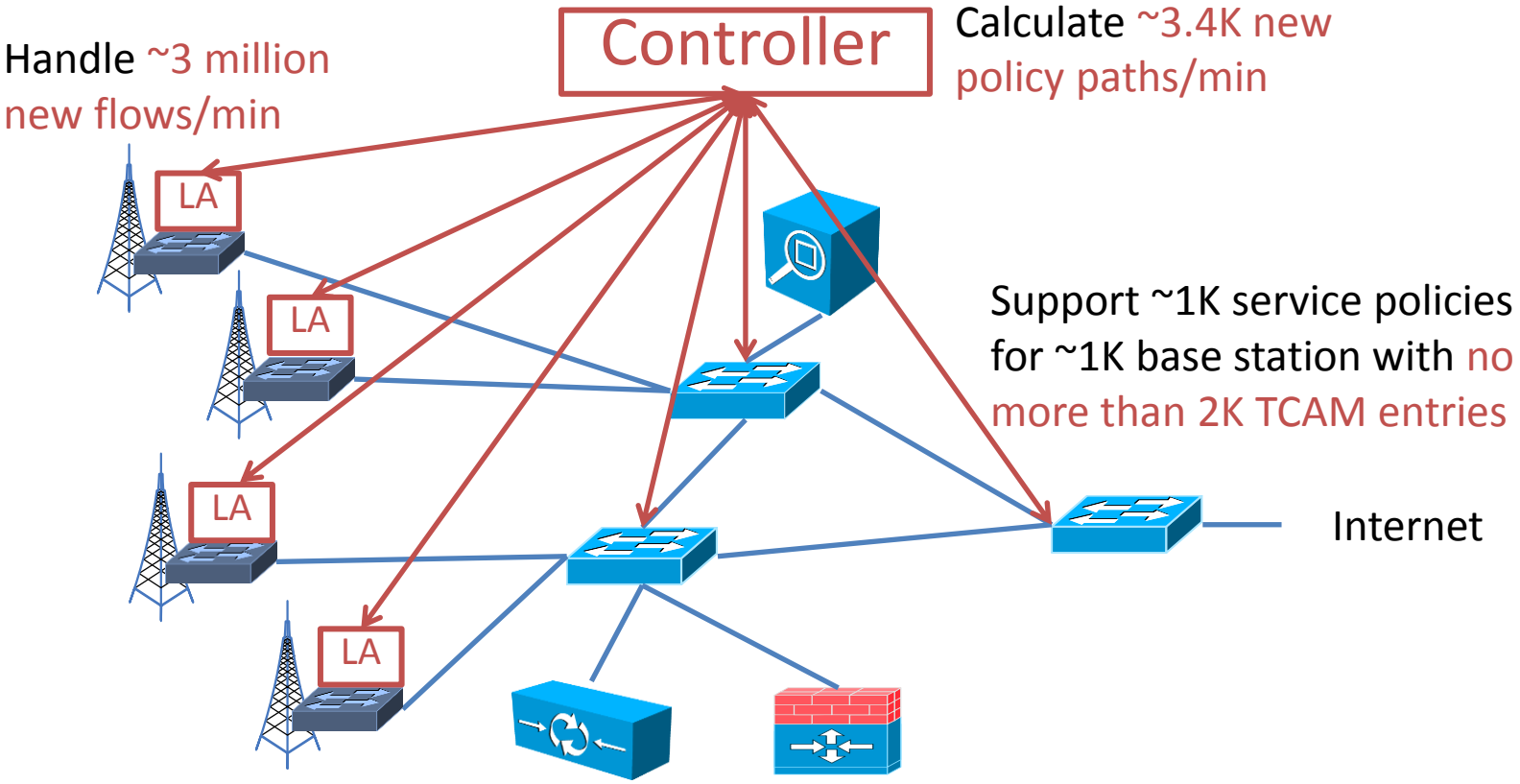
Hierarchical Controller

- Local agent (LA) at each base station
- Offload packet classification to local agents



Implementation & Evaluation

- Prototype based on Floodlight



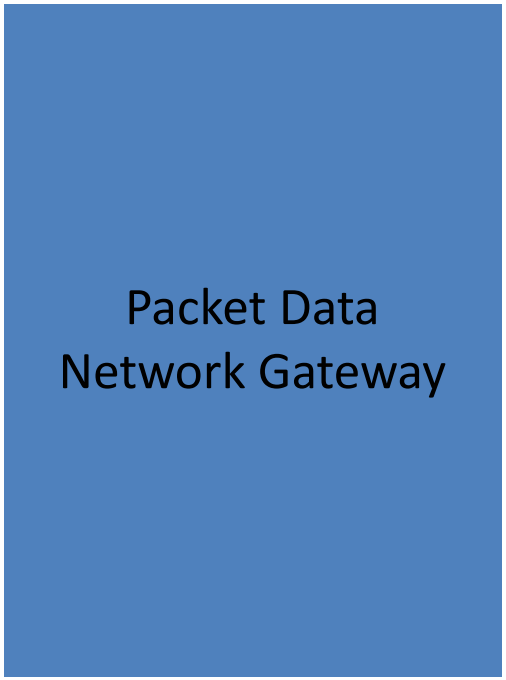
Conclusion

- CellSDN uses commodity switches and middelboxes to build flexible and cost-effective cellular core networks
- CellSDN supports fine-grained service policies and traffic management policies
- CellSDN achieves scalability with
  - Novel asymmetric edge design
  - Novel selectively multi-dimensional aggregation
  - Novel hierarchical controller design

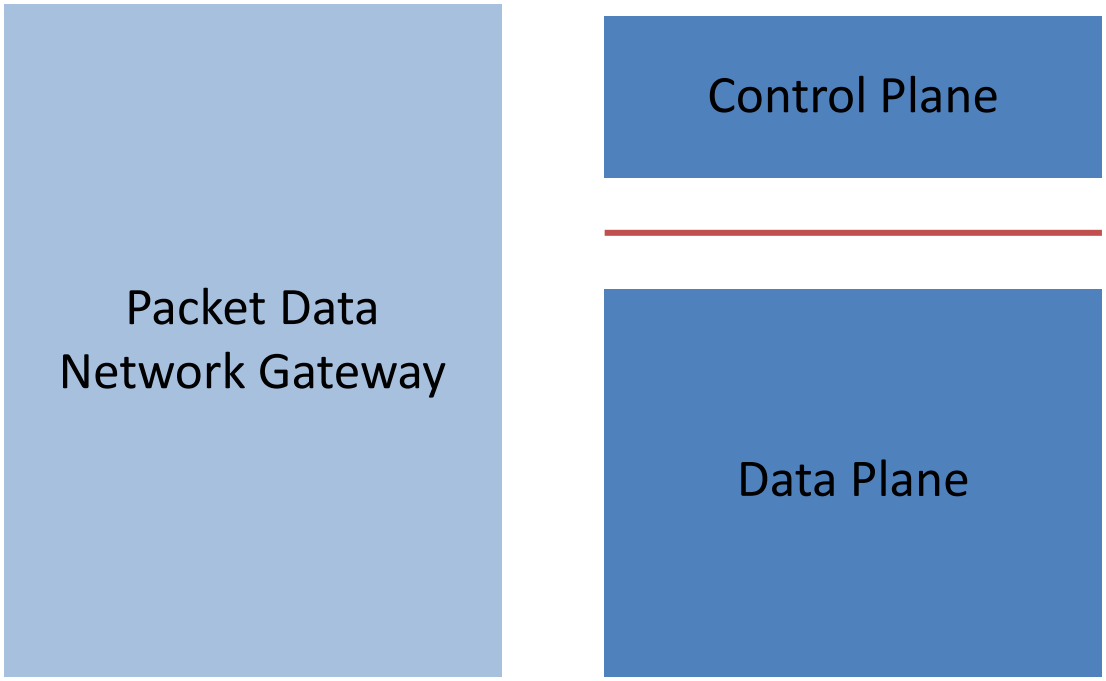
Thanks!

Contact xinjin@cs.princeton.edu
for more information!

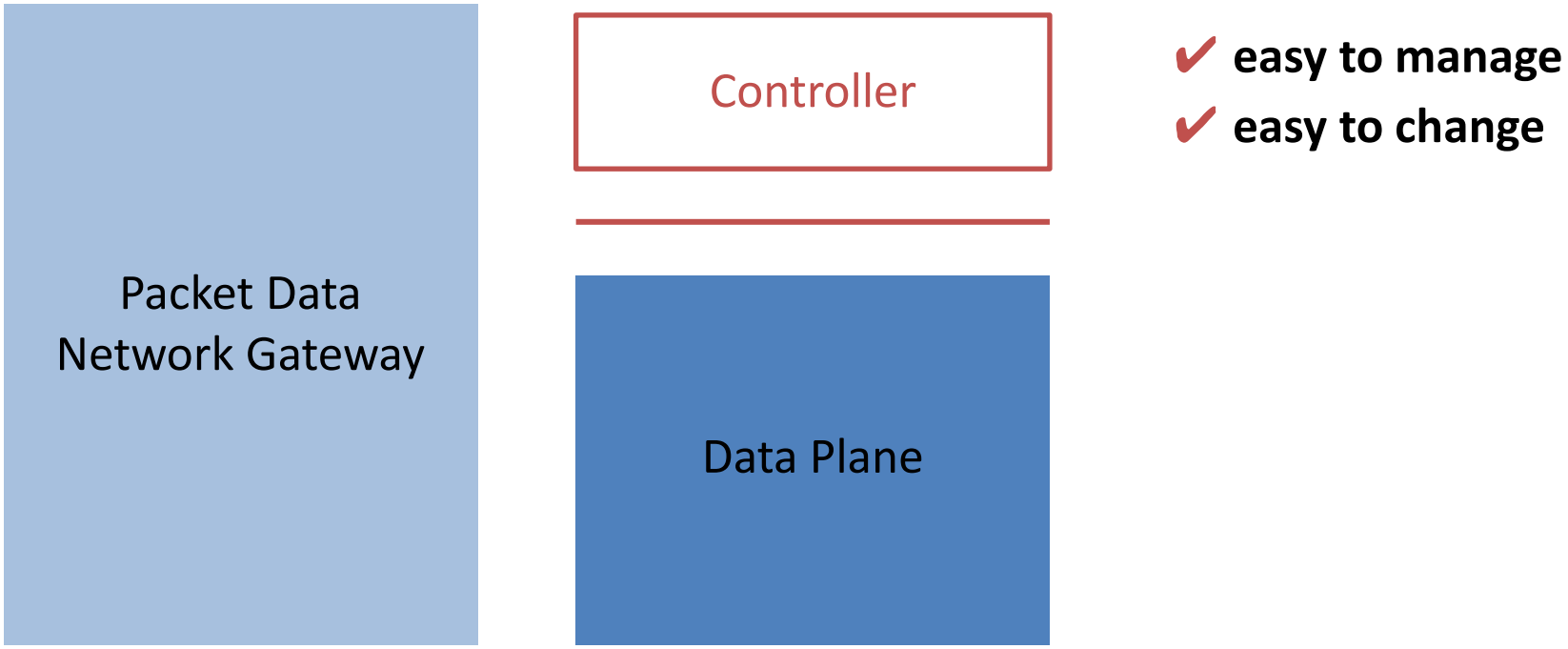
CellSDN separates control plane from data plane



CellSDN separates control plane from data plane



CellSDN separates control plane from data plane

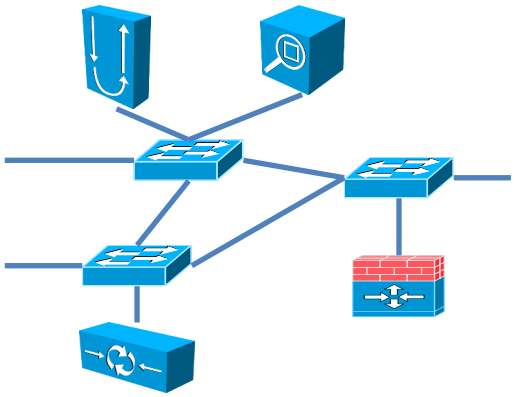


✔ **easy to manage**
✔ **easy to change**

CellSDN distributes data plane functionality to commodity switches and middleboxes

Packet Data Network Gateway

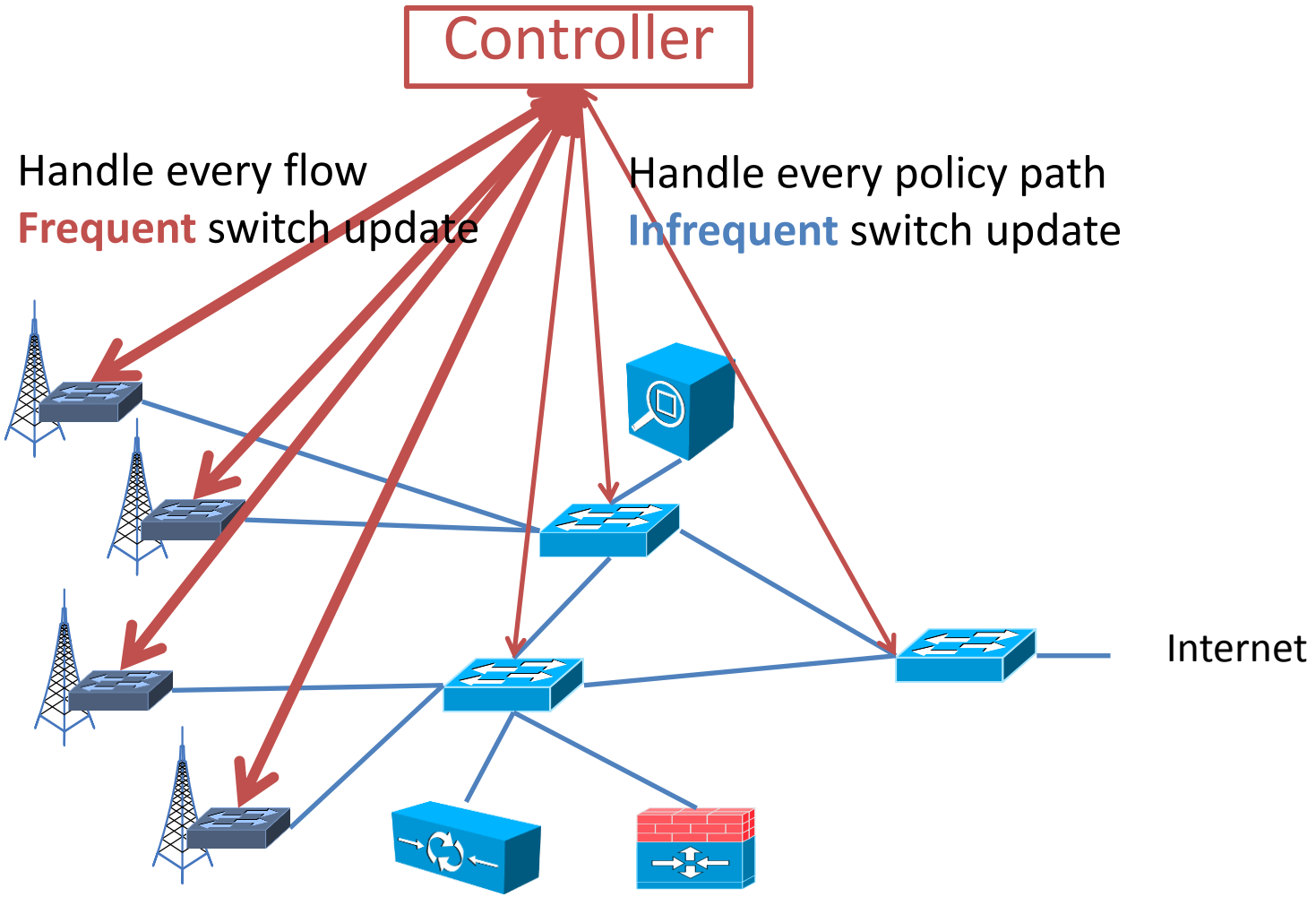
Controller



- ✔ **easy to manage**
- ✔ **easy to change**

- ✔ **mix-match**
- ✔ **up-/down-scale**
- ✔ **cheap**

Control Plane Load



Solve the problem

Step 1: Packet classification based on service policy

Service Policy

....
video traffic + gold plan customer:
firewall ➔ video transcoder
....

firewall ➔ video transcoder

Packet classification when packets arrive at the network edge