NEC Laboratories America
Relentless passion for innovation

Scaling the LTE Control-Plane for Future Mobile Access

Speaker: Rajesh Mahindra
Mobile Communications & Networking
NEC Labs America

Other Authors: Arijit Banerjee, Utah University
Karthik Sundaresan, NEC Labs
Sneha Kasera, Utah University
Kobus Van der Merwe, Utah University
Sampath Rangarajan, NEC Labs

Part 1: Motivation & Background

Trends

1. Control signaling storm in Mobile Networks:
   - Growth in the signaling traffic 50% faster than the growth in data traffic.
   - 290000 control messages/sec for 1 million users!
   - In a European network, about 2500 signals per hour were generated by a single application causing network outages.

- Always-on Connectivity and Cloud Computing
- Explosion of IoT devices (Internet of Things): Projected at 26 Billion by 2020
- Conservation of battery: Transition to idle mode

2. Adoption of NFV in LTE:
   - 5G vision for RAN: explore higher frequencies (e.g., mmWave)
   - 5G vision for Core Network: Virtualization and Cloudification
     - » Increased flexibility and customizability in deployment and operation
     - » Reduced costs and procurement delays

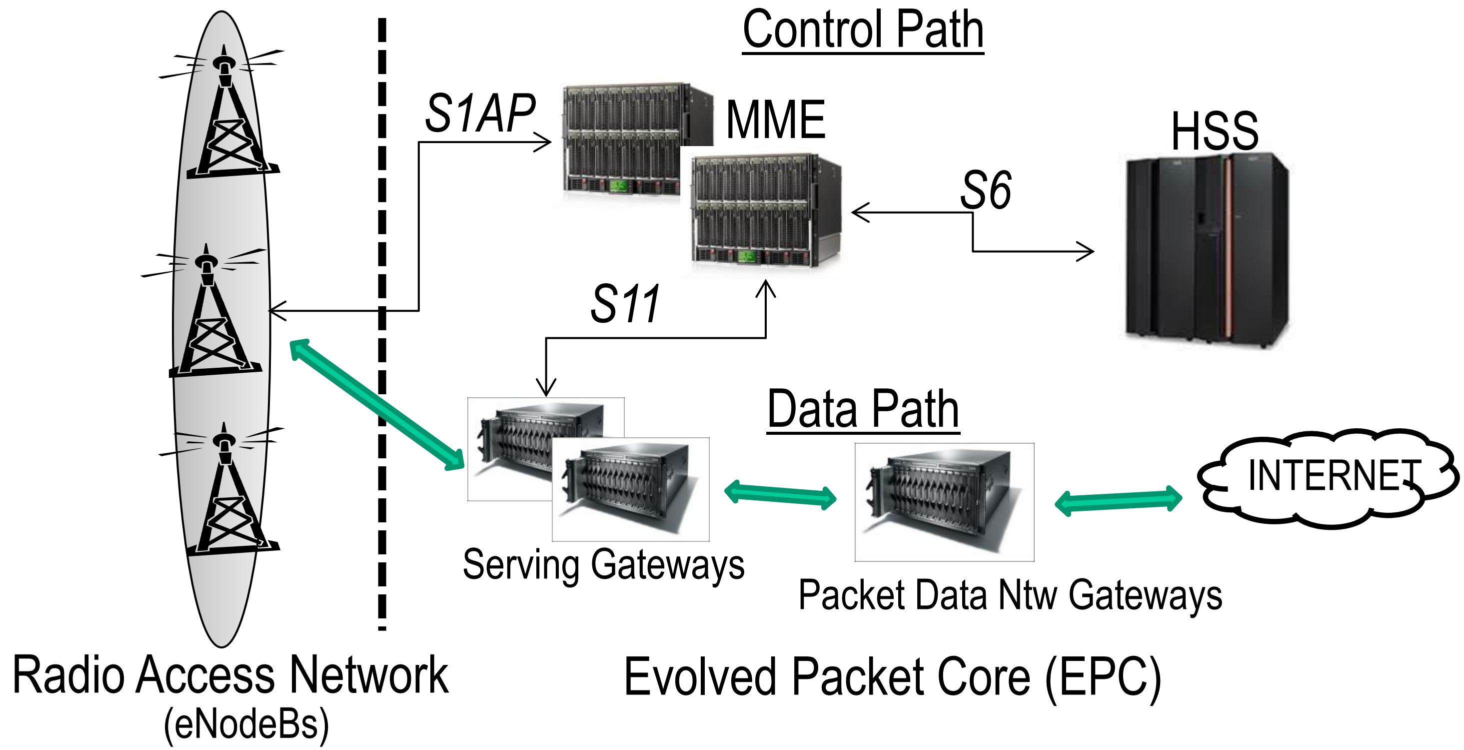
Problem Statement

Goal: Effective Virtualization of the LTE Control-plane

- In LTE, the main control-plane entity is the MME (Mobility Management Entity)
  - The MME processes 5 times more signaling than any other entity
  - Execute MME functionality on a cluster of Virtual Machines (VMs)

- Effective virtualization of MME includes:
  - Performance: Overloaded MMEs directly affect user experience:
    - Idle-Active transition delays cause connectivity delays
    - Handover delays effect TCP performance
  - Cost-effectiveness: Control-signaling does not generate direct revenue:
    - Over-provisioning: Under-utilized VMs
    - Under-provisioning: Processing delays

Background: LTE Networks



MME Virtualization Requirements

- Elasticity of compute resources:
  - VMs are scaled-in and out dynamically with expected load
  - Proactive approaches to ensure efficient load balancing
    - Lower processing delays for a given set of VMs OR
    - Reduced number of VMs to meet specific delay requirements
- Scale Of Operation:
  - Typically, number of active devices (that generate signaling) << total number of registered devices
  - Expected to be more pronounced with IoT devices
- 3GPP Compatibility:
  - Ensures easy and incremental deployment

Part 2: State of the Art

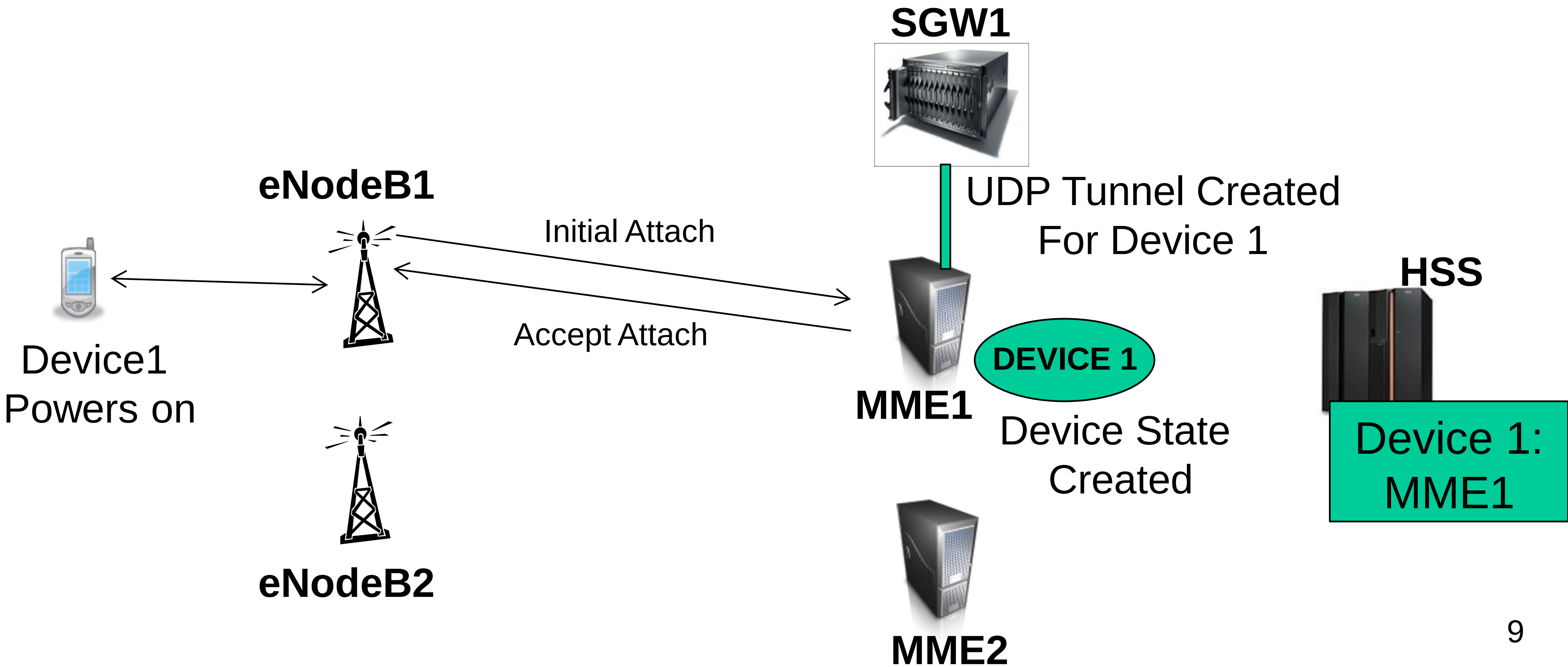
Today's MME Implementations

NEC Laboratories America
Relentless passion for innovation

- Current implementations are ill-suited for virtualized MMEs:
  - hardware-based MME architecture
  - Porting code to VMs is highly inefficient
- Fundamental Problem:
  - Static Assignment of devices to MMEs
  - Persistent sessions per device with Serving gateways, HSSs and eNodeBs/devices

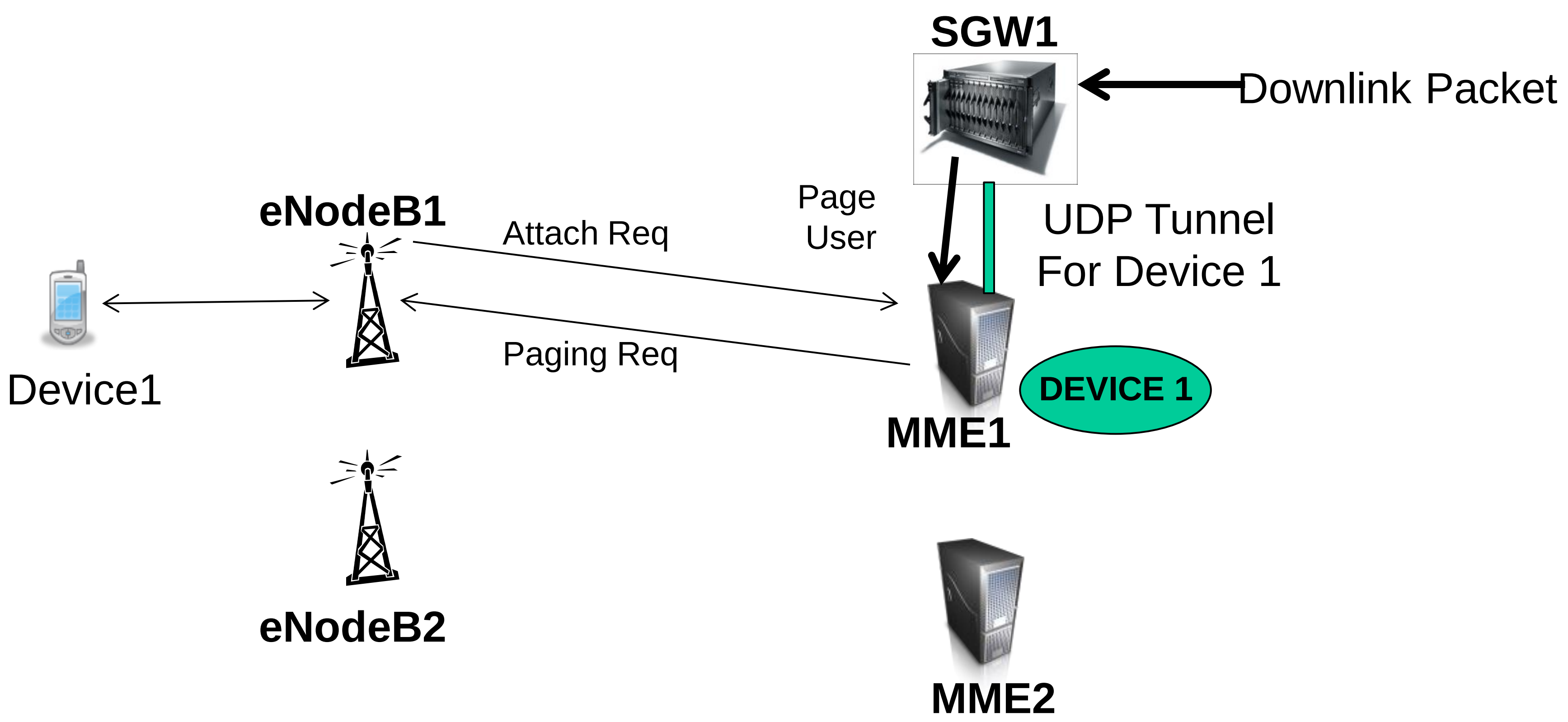
Today's MME Implementations

- Once registered, a device is persistently assigned to an MME
  – The device, its assigned Serving Gateway (S-GW) and the HSS store the MME address and;
  – Subsequent control messages from the device, SGW and HSS are sent to the same MME.

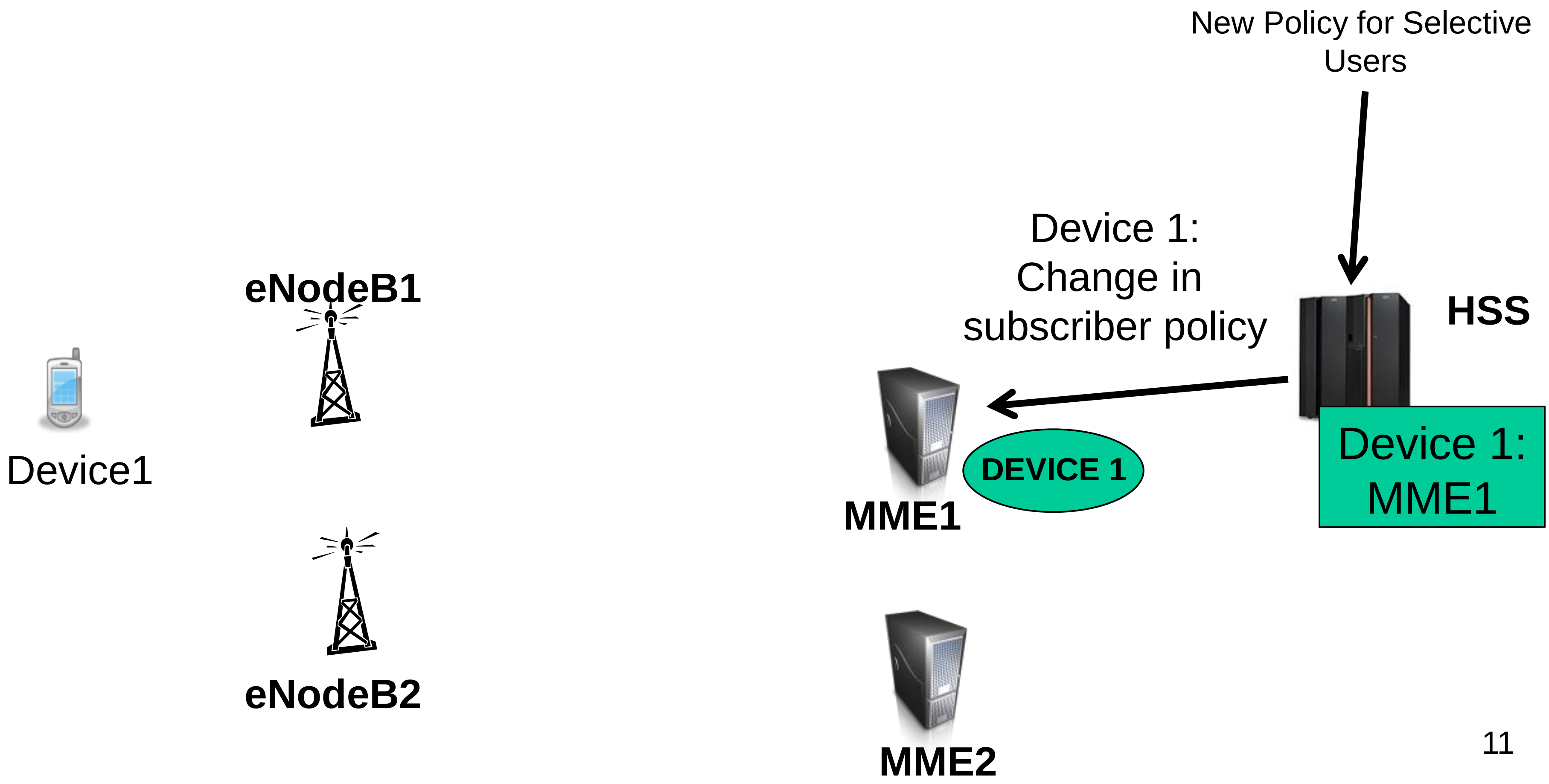


Today's MME Implementations

NEC Laboratories America
Relentless passion for innovation



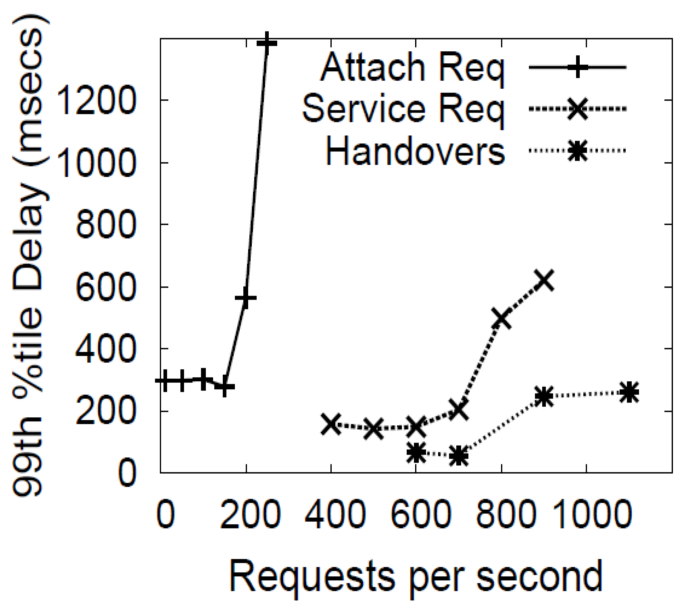
Today's MME Implementations



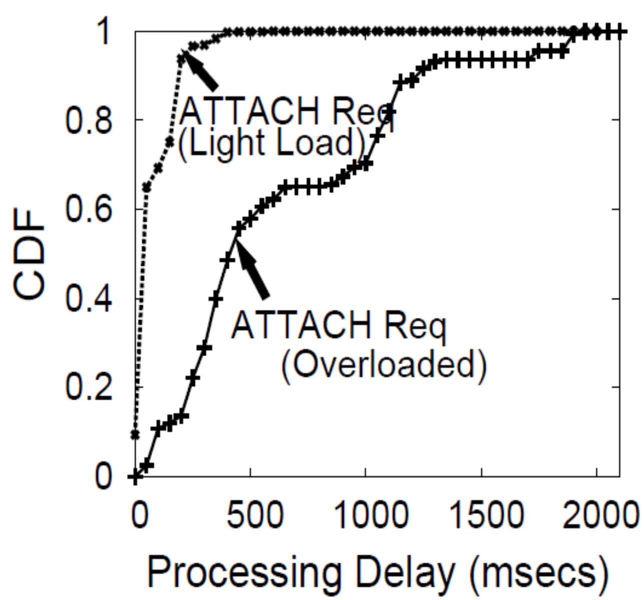
Limitations of Current Implementations

Static Configurations result in inefficiency and inflexibility

1. *Elasticity*: Only new (unregistered) devices can be assigned to new VMs
2. *Load-balancing*: Re-assignment of device to a new MME requires control messages to the device, SGW and HSS



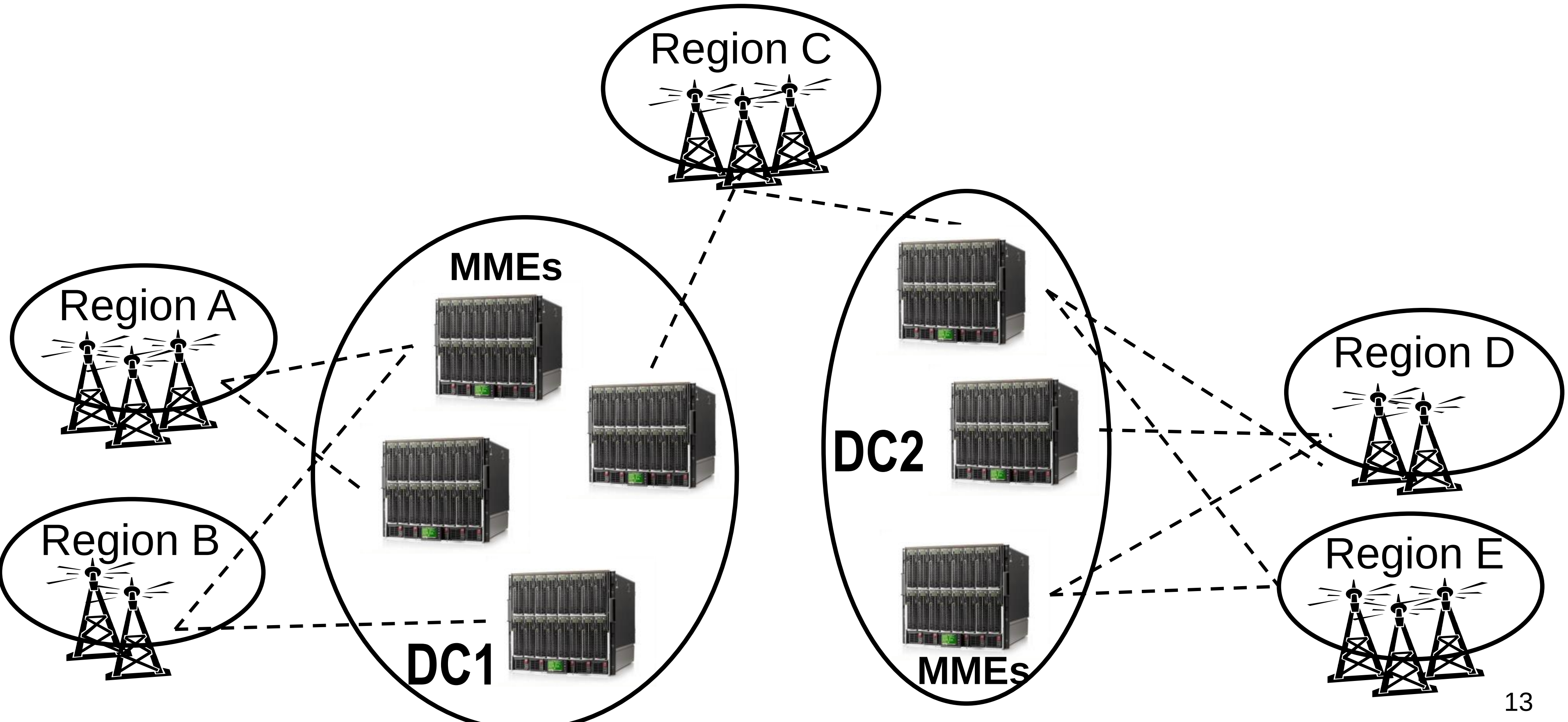
(a) Static Assignment



(b) Overload Protection

Limitations of Current Implementations

3. *Geo-multiplexing across DCs*: Inflexibility to perform fine-grained load balancing across MME VMs in different DCs



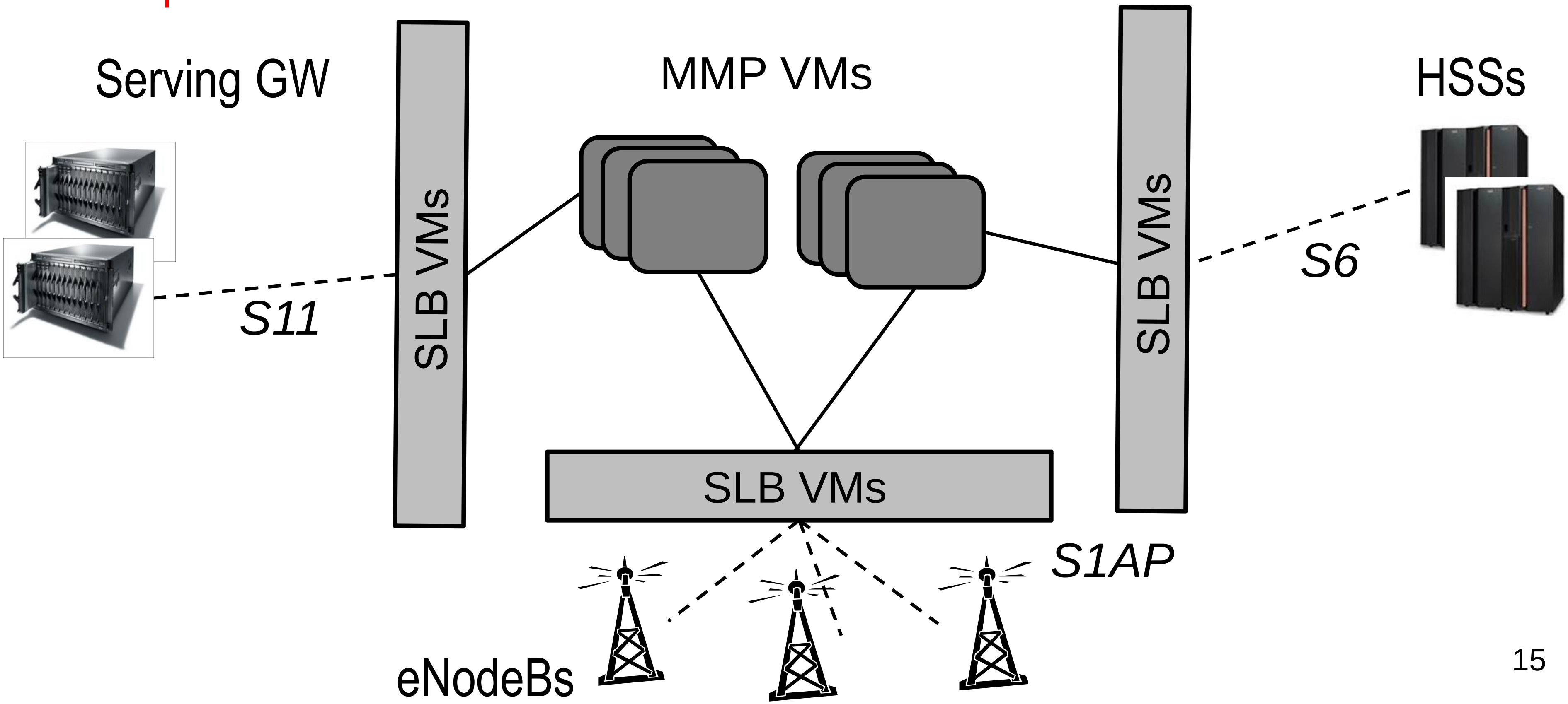
Part 3: Design Overview

Design Architecture

NEC Laboratories America
Relentless passion for innovation

Decouple standard interfaces from MME Device management:

1. SLB: Load-balancers that forward requests from devices, SGW and HSS to the appropriate MMP VM
2. MMP: MME Processing entitles that store device state and process device requests.
   - MMP VMs exchange device states to ensure re-assignment during scaling procedures



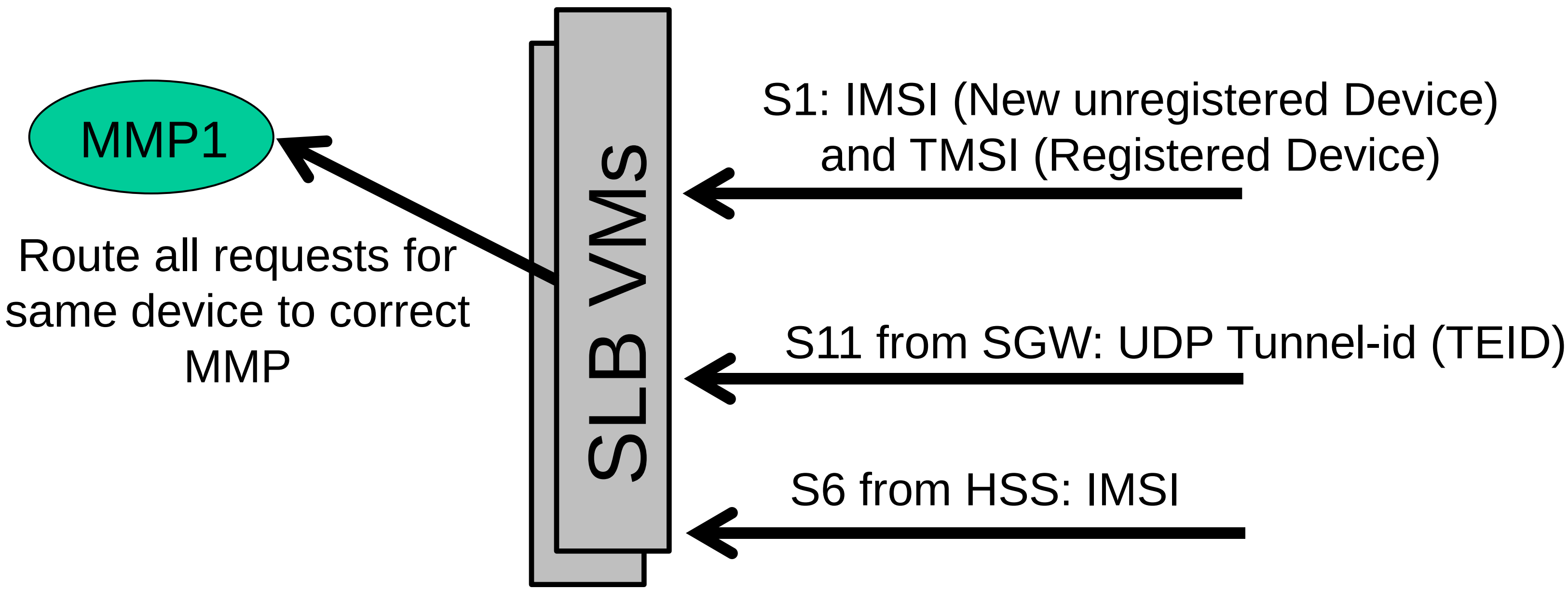
Design Considerations

NEC Laboratories America
Relentless passion for innovation

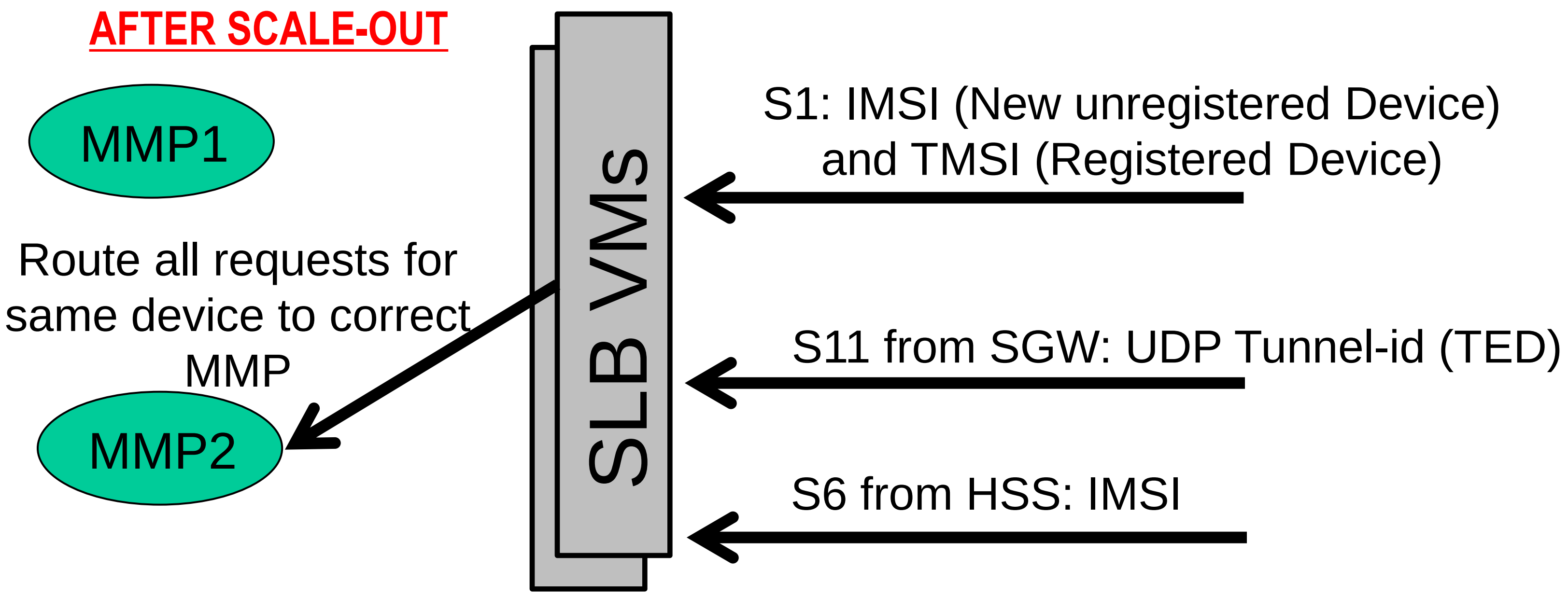
How do we dynamically (re)-assign devices to MMP VMs as the VMs are scaled-in and out?

- Scalable with the expected surge in the number of devices
- Ensure efficient load balancing without over-provisioning
- SLB/Routing bottlenecks:
  - Multiple SLB VMs may have to route the same device requests
  - Each interface contains different ids or keys for routing
    - SLB VMs will need to maintain separate table to route the requests from each interface

Design Considerations



Design Considerations



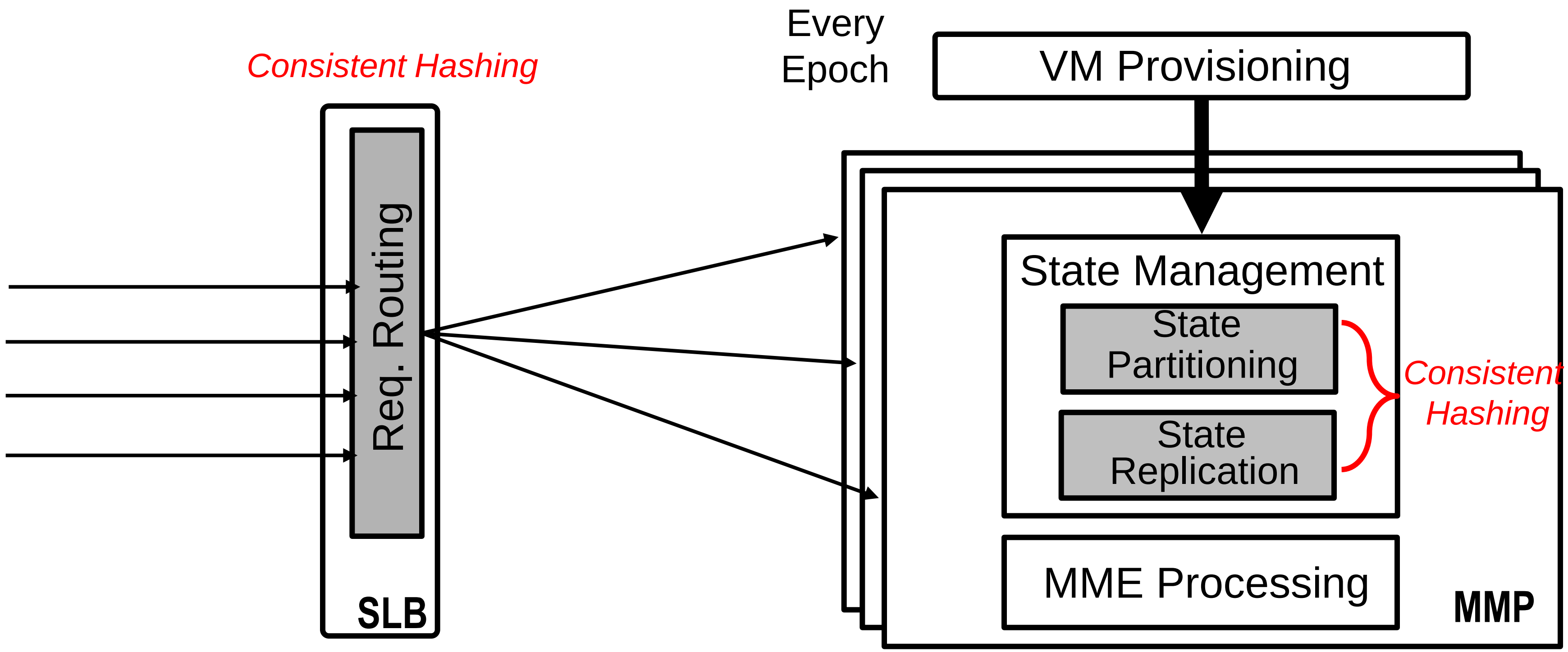
Our Approach: SCALE

- Leverage concept from distributed data-stores:
  - *Consistent Hashing* (e.g., Amazon DynamoDB and Facebook Cassandara)
    - Provably practical at scale
  - *Replicate* device state across multiple MMP VMs
    - fine-grained load balancing

- Apply it within the context of virtual MMEs
  - Coupled provisioning for computation of device requests and storage of device state
  - Replication of device state is costly, requiring tight synchronization

SCALE Components

- VM Provisioning: Every hour(s), decides when to instantiate a new VM (scale out) or bring down an existing VM(scale in)
- State Partitioning: (Re)-distribution of state across existing MMP VMs
  State Replication: Copies device state across MMP VMs to ensure efficient load-balancing



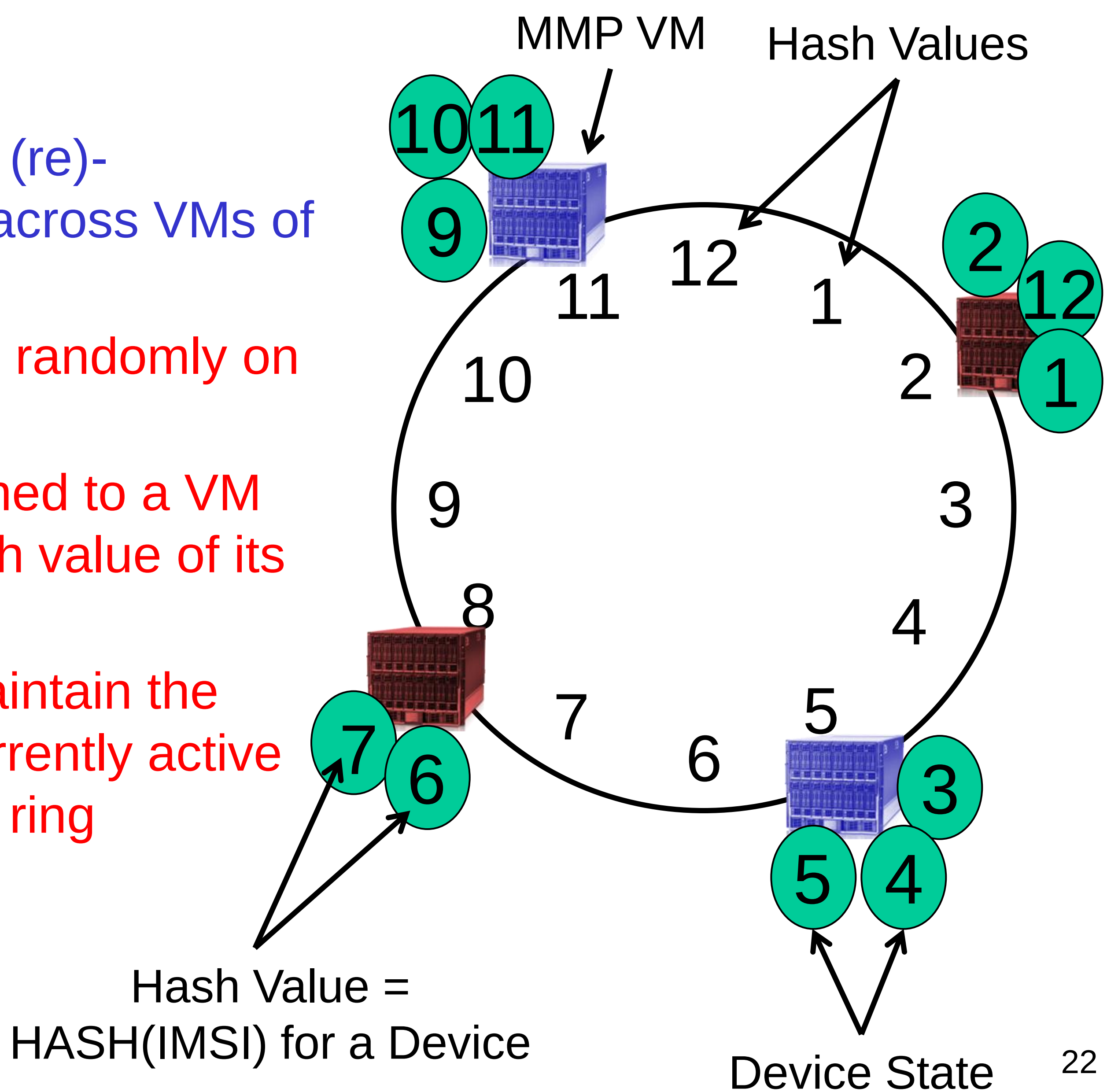
Part 4(a): Design within a single DC

How is consistent hashing applied?

NEC Laboratories America
Relentless passion for innovation

Scalable, decentralized (re)-assignment of devices across VMs of a single DC

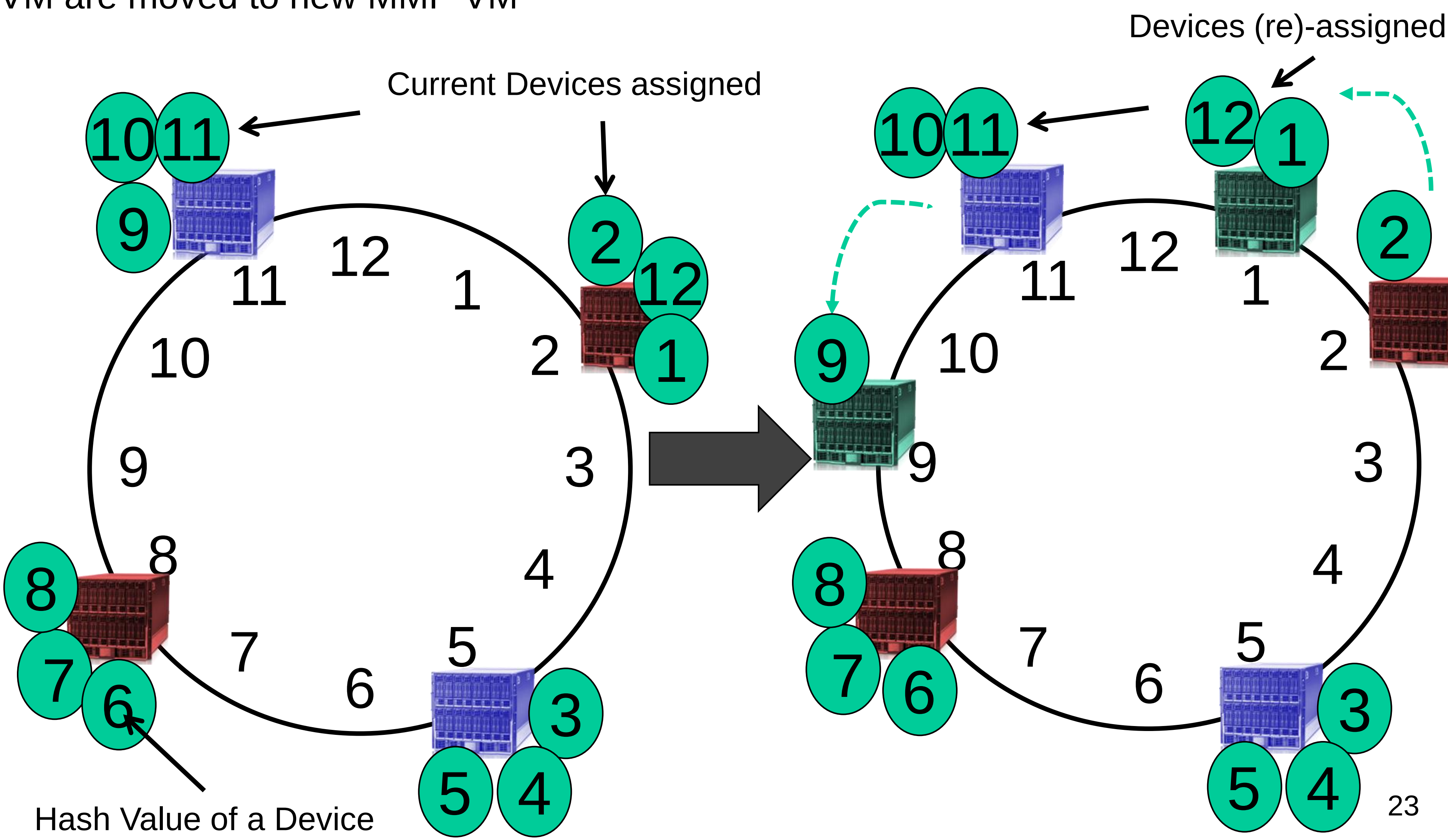
- MMPs are placed randomly on a hash ring
- A device is assigned to a VM based on the hash value of its IMSI
- SLB VMs only maintain the location of the currently active MMP VMs on the ring



Scale-out procedure (Scale-in is similar)

<u>Step 1:</u> The new MMP VM is randomly placed on the ring

<u>Step 2:</u> The state of Devices of current MMP VMs that fall in the range of the new MMP VM are moved to new MMP VM



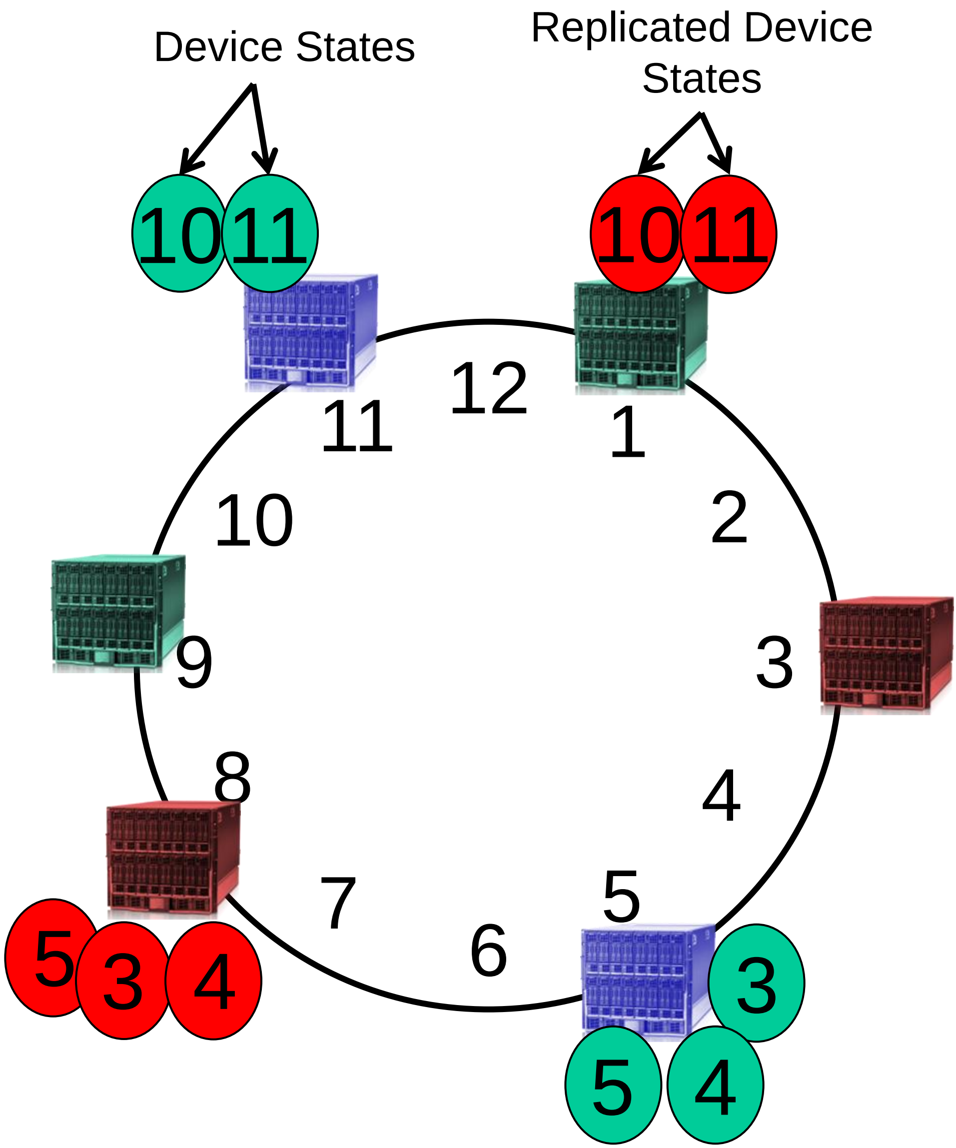
Proactive Replication: Efficient Load Balancing

NEC Laboratories America
Relentless passion for innovation

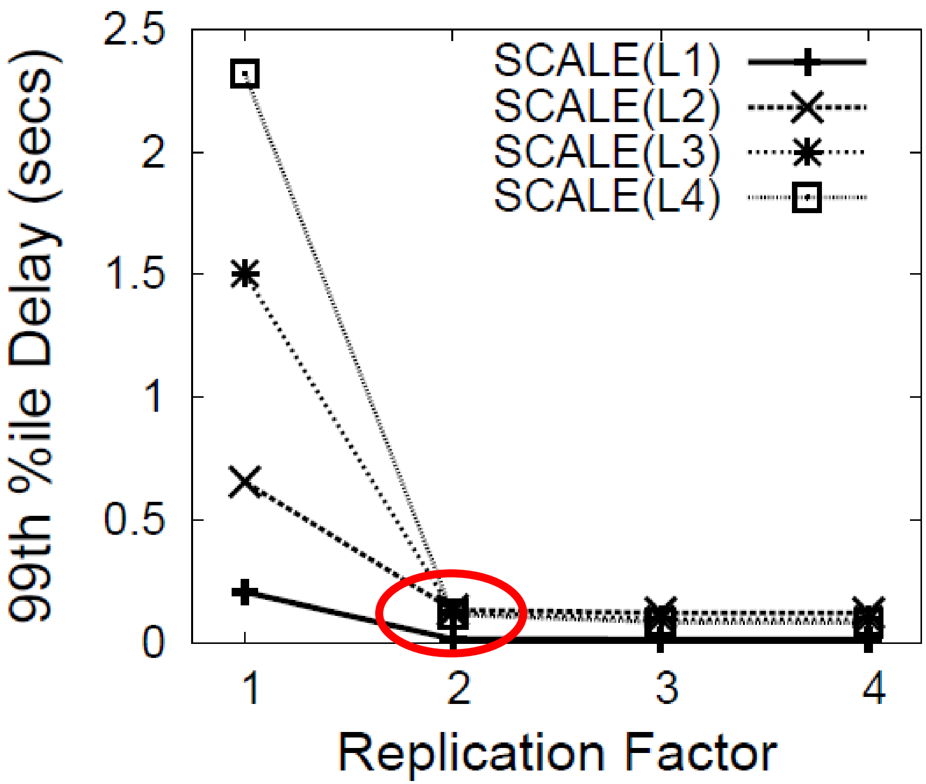
1. Each MMP VM is placed as multiple tokens on the ring
2. The device state assigned to a token of the MMP VM, is replicated to the adjacent token of another MMP VM

Leveraging hashing for replication ensures no additional overhead for SLB VMs:

- In real-time, the SLB VMs forward the request of a device to the least loaded MMP VM



Proactive Replication: Efficient Load Balancing



We derived an analytical model and performed extensive simulations to show that:

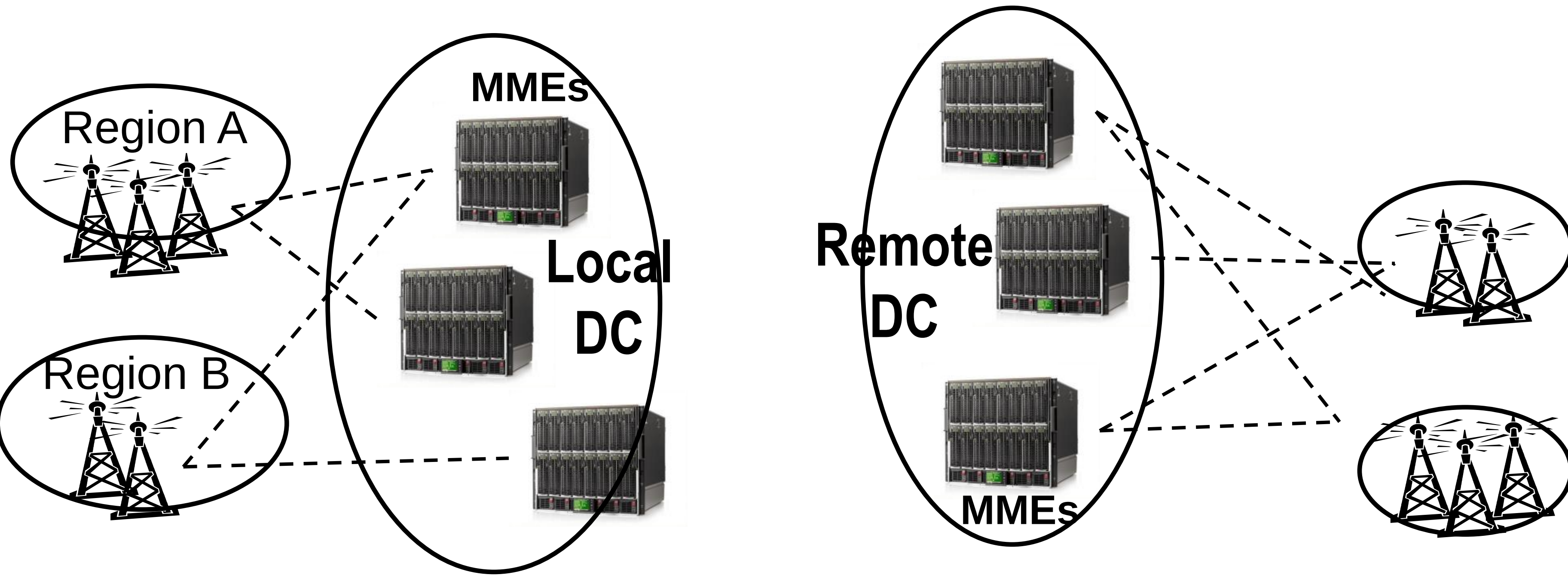
Our procedure of consistent hashing + replication results in efficient load-balancing with <u>only 2 copies of device state</u>

L1-L4: Increasing levels of Load Skewness across the MMP VMs

Part 4(b): Design across DCs

Proactive Replication Across DCs

- SCALE replicates a device state in an additional MMP VM in the local DC
- SCALE also replicates the state of certain devices to MMP VMs at remote DCs
  - Enables fine-grained load balancing across DCs
  - SCALE replicates devices at remote DC to minimize latency



Proactive Replication Across DCs

- Selection of Device: Medium activity pattern
  - Highly active devices are only assigned at the local DC to reduce average latencies
  - Replicating highly dormant devices to remote DC does not help load balancing
- Selection of remote DC: Selection is probabilistic based on the metric ‘p’:

$$p = \frac{\frac{1}{D_{ik}}}{\sum_{i=1}^{C} \frac{1}{D_{ij}}}$$

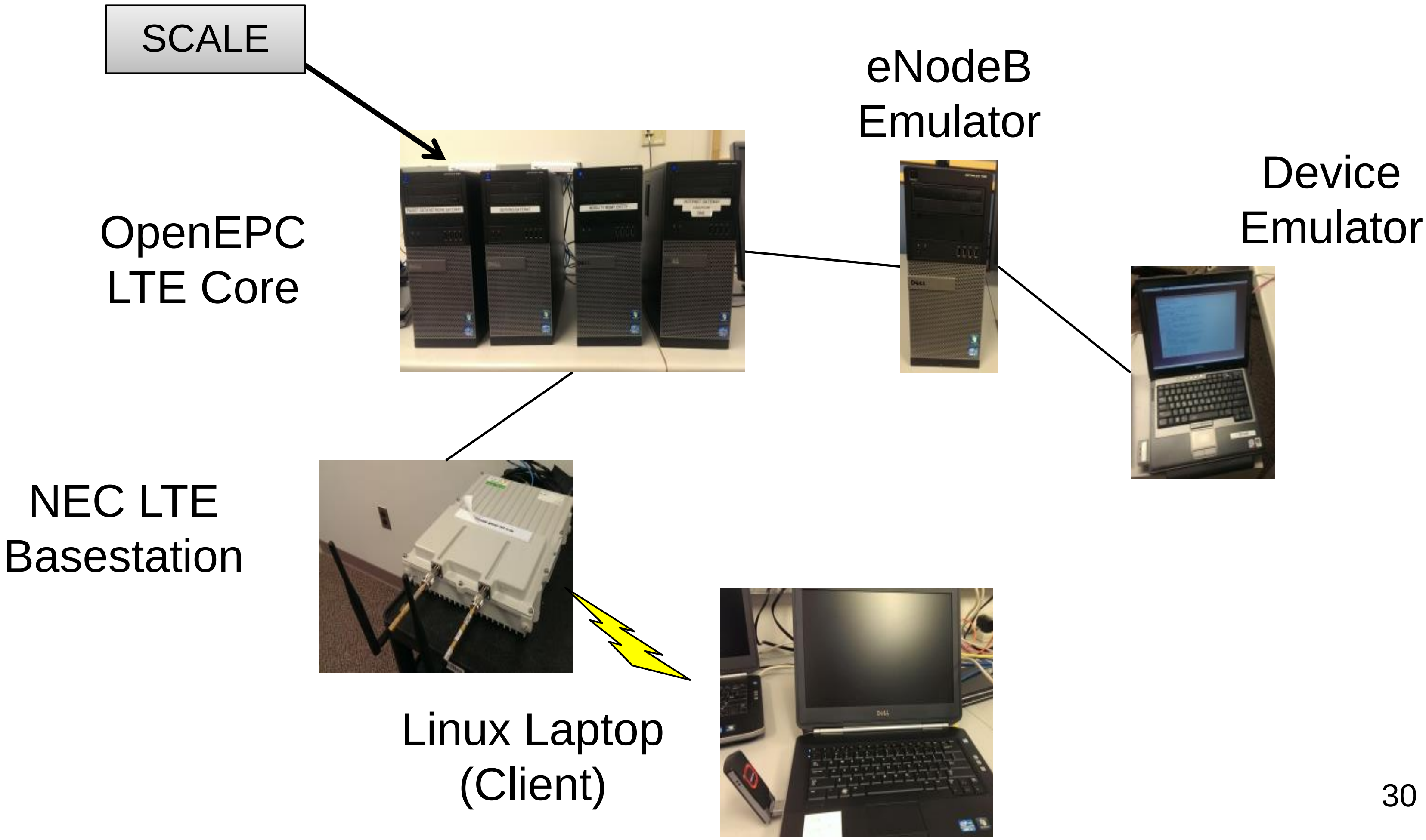
where D_{ij} is the propagation delay between DC i and j;

- In real-time, the SLB VM always forwards the request of a device to the least loaded MMP VM in the local DC
  - If overloaded, the local MMP VM forwards the request to the MMP VM in the remote DC

Part 5: Prototype & Evaluation

Prototype

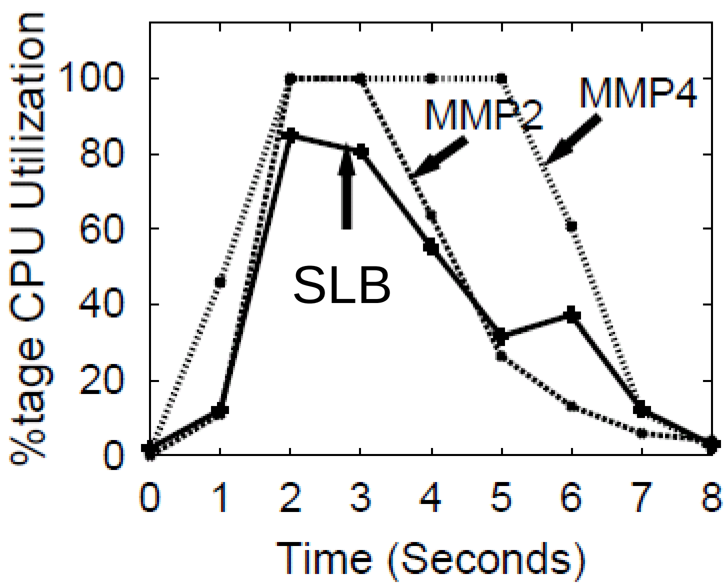
- The OpenEPC testbed is a C (linux) based Release 9 LTE network
- SCALE is implemented within the openEPC codebase
- Implementation effort includes splitting the MME into SLB and MMPs



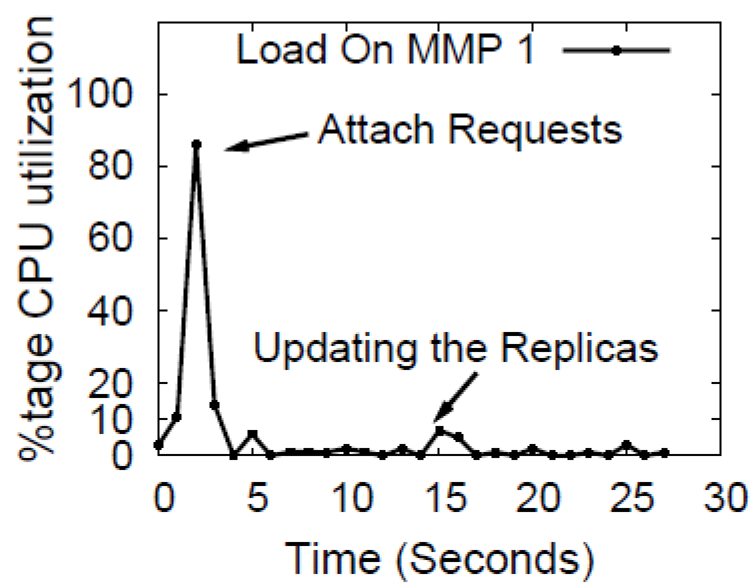
Benchmarking Experiments

- Expt1 SLB Overhead: Current prototype supports 5 MMP VMs for a single SLB VM at full load

- Expt2 Replication Overhead: The overhead of synchronizing device state (copying) is less than 8%



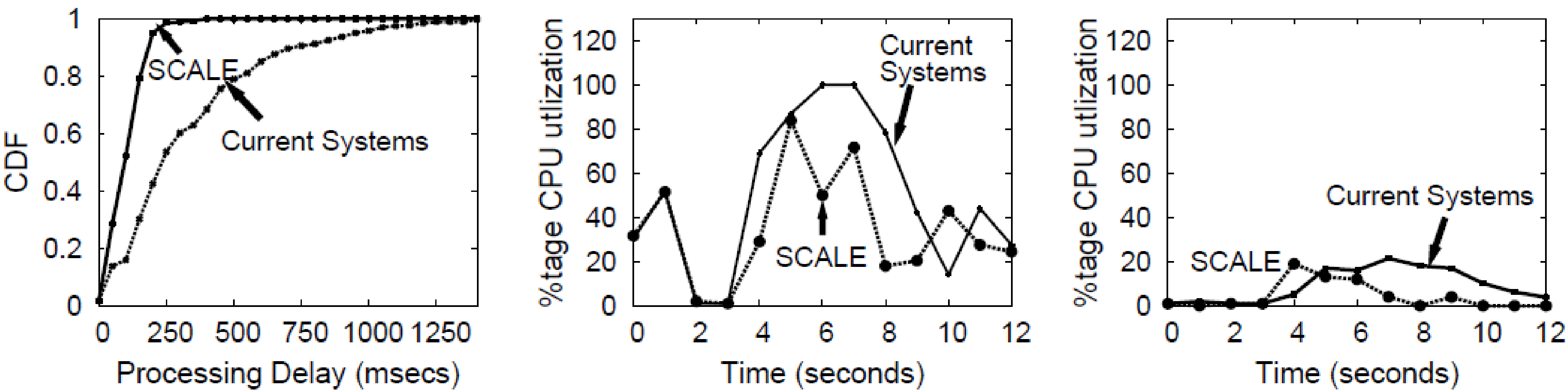
(a) SLB Processing



(b) State Replication

Efficacy of SCALE compared to current implementations

- SCALE performs proactive replication vs reactive replication in current MME systems:
  - (a) SCALE ensures lower control-plane processing delays
  - (b) & (c) SCALE ensures lower CPU loads since it does not involve per-device overheads to re-assign devices



(a) Processing delays (b) Load on MMP1 (c) Load on MMP2

NEC Laboratories America
Relentless passion for innovation

Conclusion

- Current MME implementations:
  - Ill-suited for virtualized environments
  - Rely on over-provisioning to avoid overload
  - Will not scale to next-generation of IoT-based mobile access
- SCALE effectively applies concepts from distributed systems to virtual MME systems:
  - Decoupling architecture enables elasticity
  - Consistent hashing ensures scalable re-distribution of devices
  - Proactive replication strategy ensures efficient load-balancing