ACM SIGCOMM 2008 Computer Communication Review

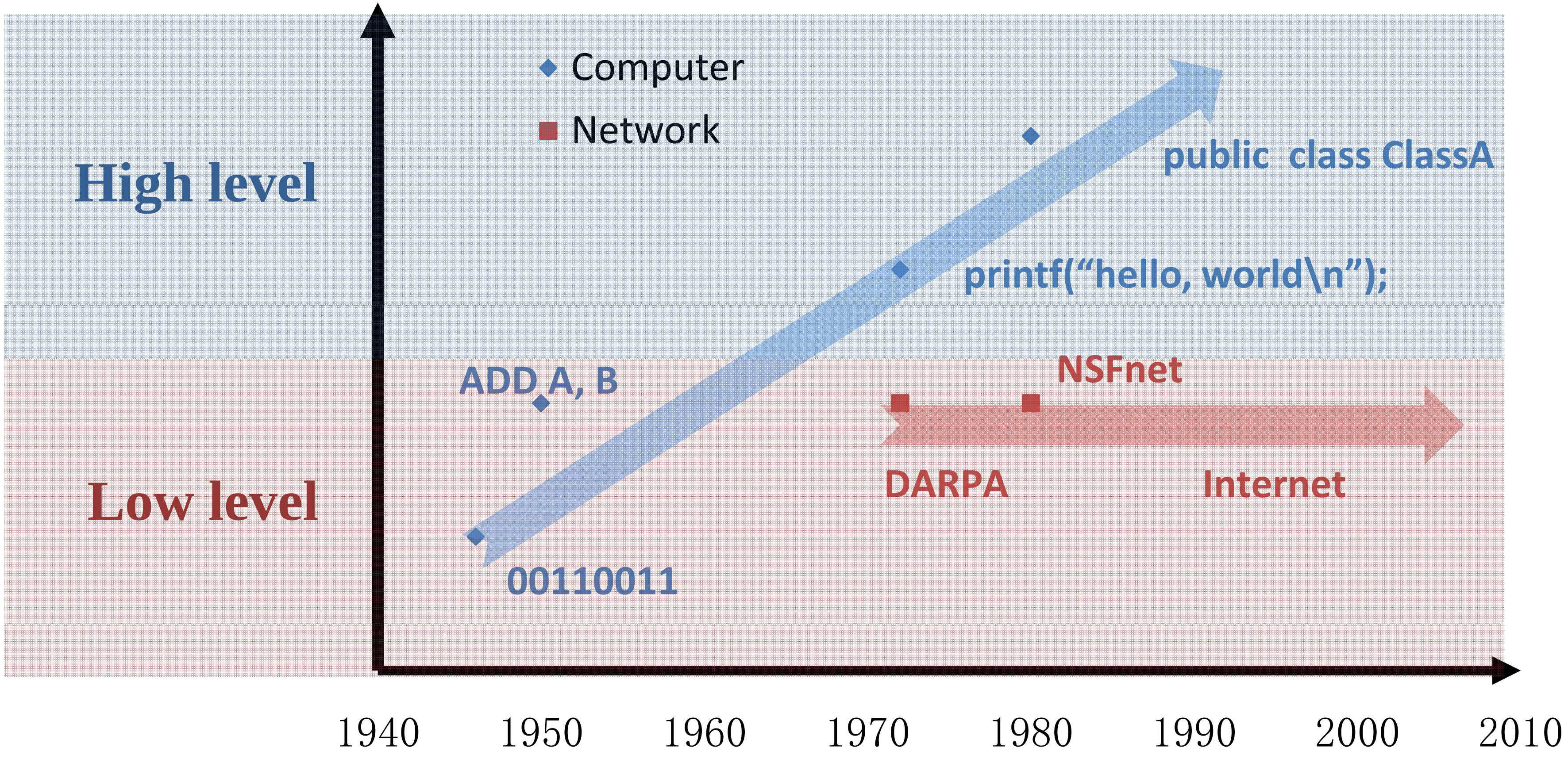
NOX: Towards an Operating System for Networks

Natasha Gude, Teemu Koponen, Justin Pettit, Ben Pfaff,
Martin Casado, Nick McKeown and Scott Shenker

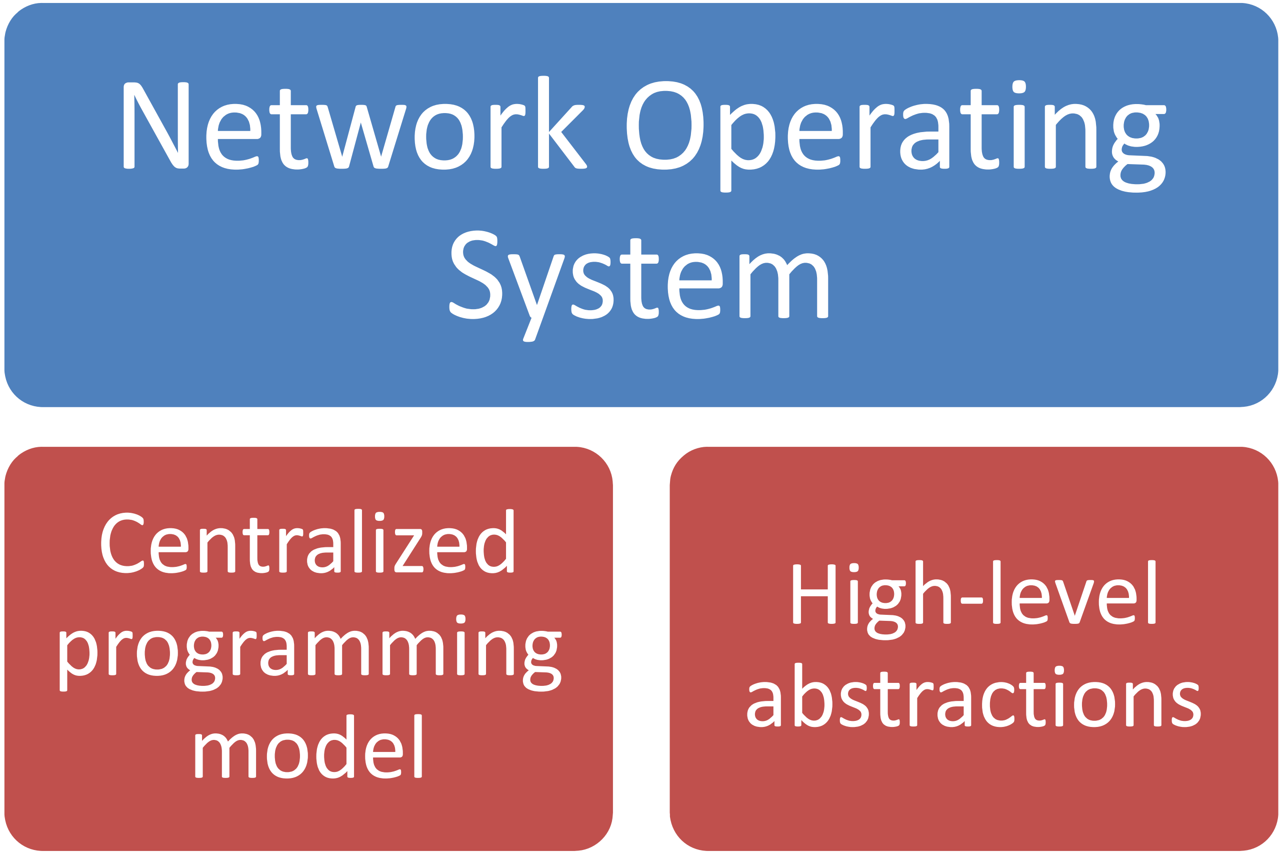
Outline

- Introduction
- NOX overview
- Programmatic Interface
- Example Applications

Computer vs Network



Two major conceptions

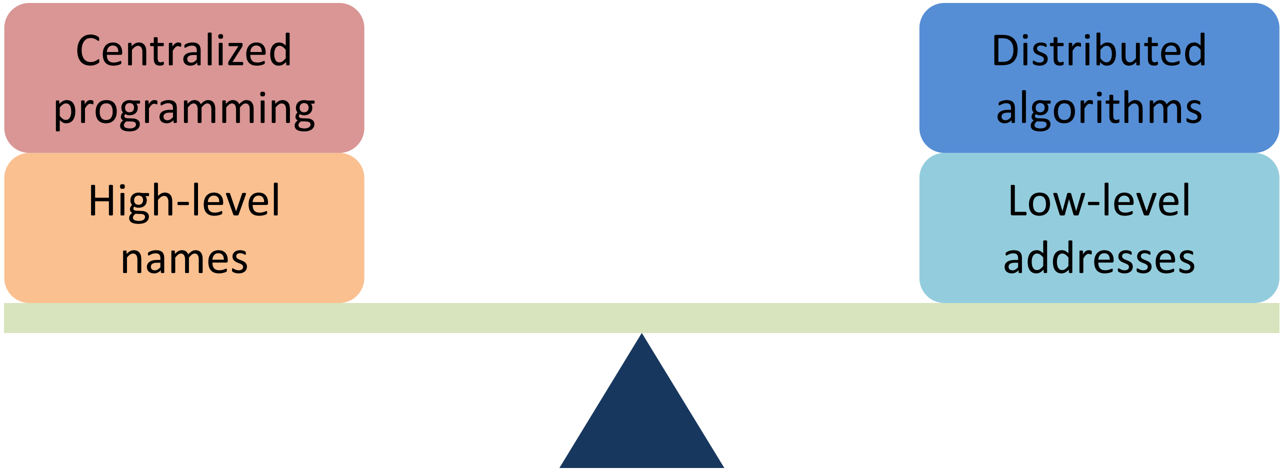


Network Operating System

Centralized programming model

High-level abstractions

Challenges



Centralized programming

High-level names

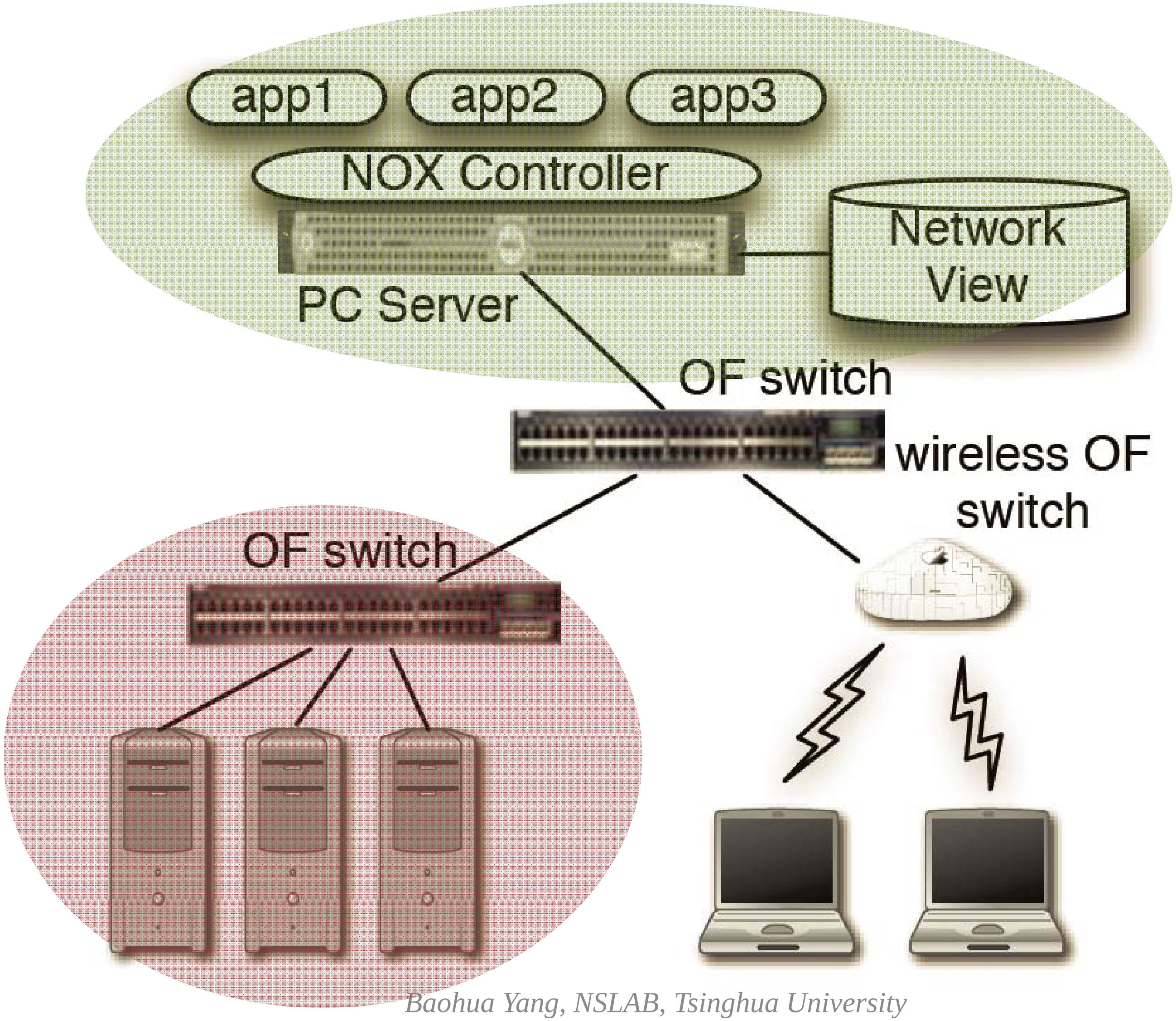
Distributed algorithms

Low-level addresses

Can one build a network operating system at significant scale?

NOX!

Components


app1
app2
app3
NOX Controller
Network View
PC Server
OF switch
wireless OF switch
OF switch

Granularity

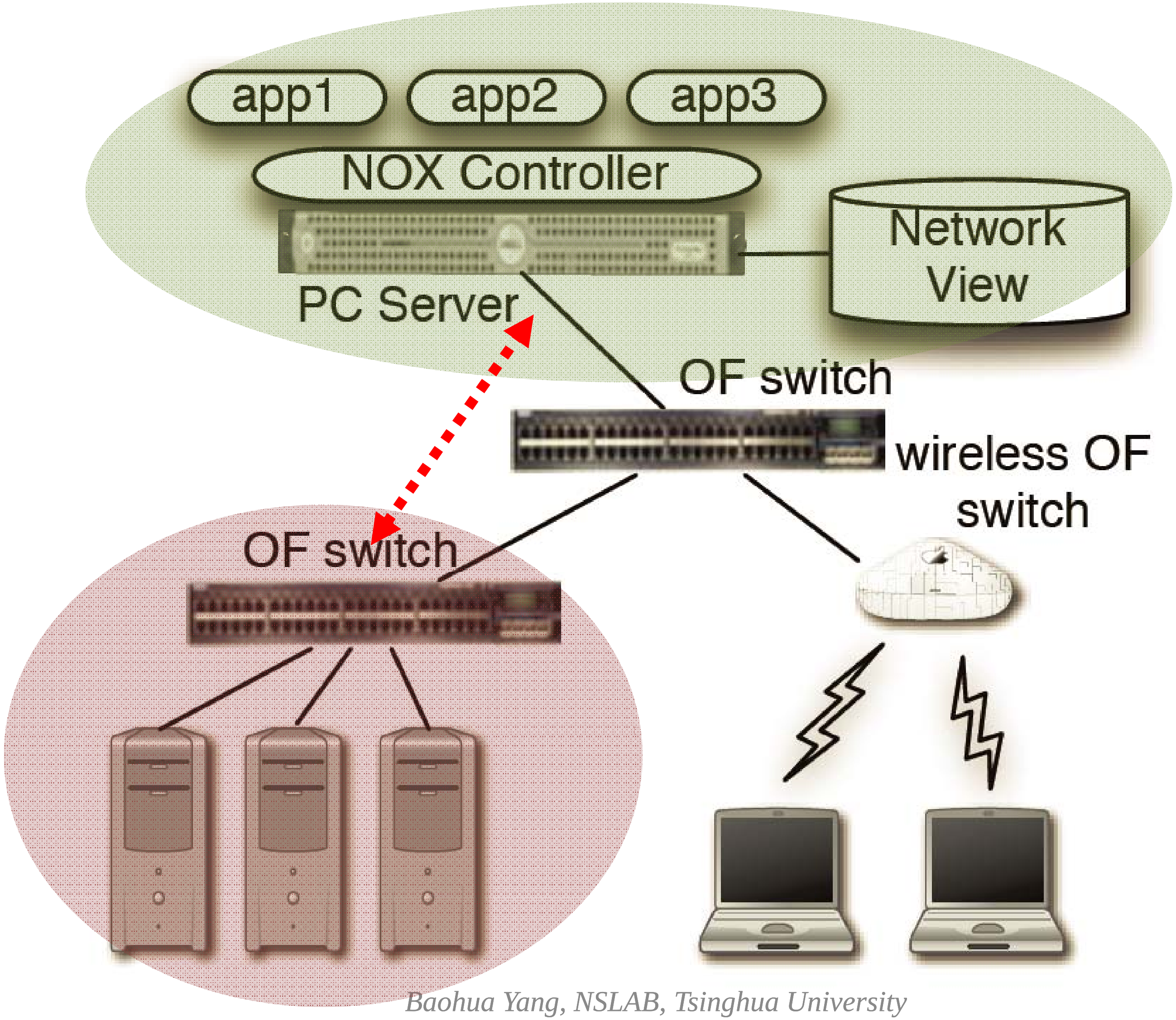
- **Observation** granularity
  - Switch-level topology
  - Locations of users, hosts, middleboxes, etc.
  - **But NO** traffic state
- **Control** granularity
  - Flows-based
- Scalability vs flexibility

Switch Abstract

- OpenFlow switches
  - Flow table form

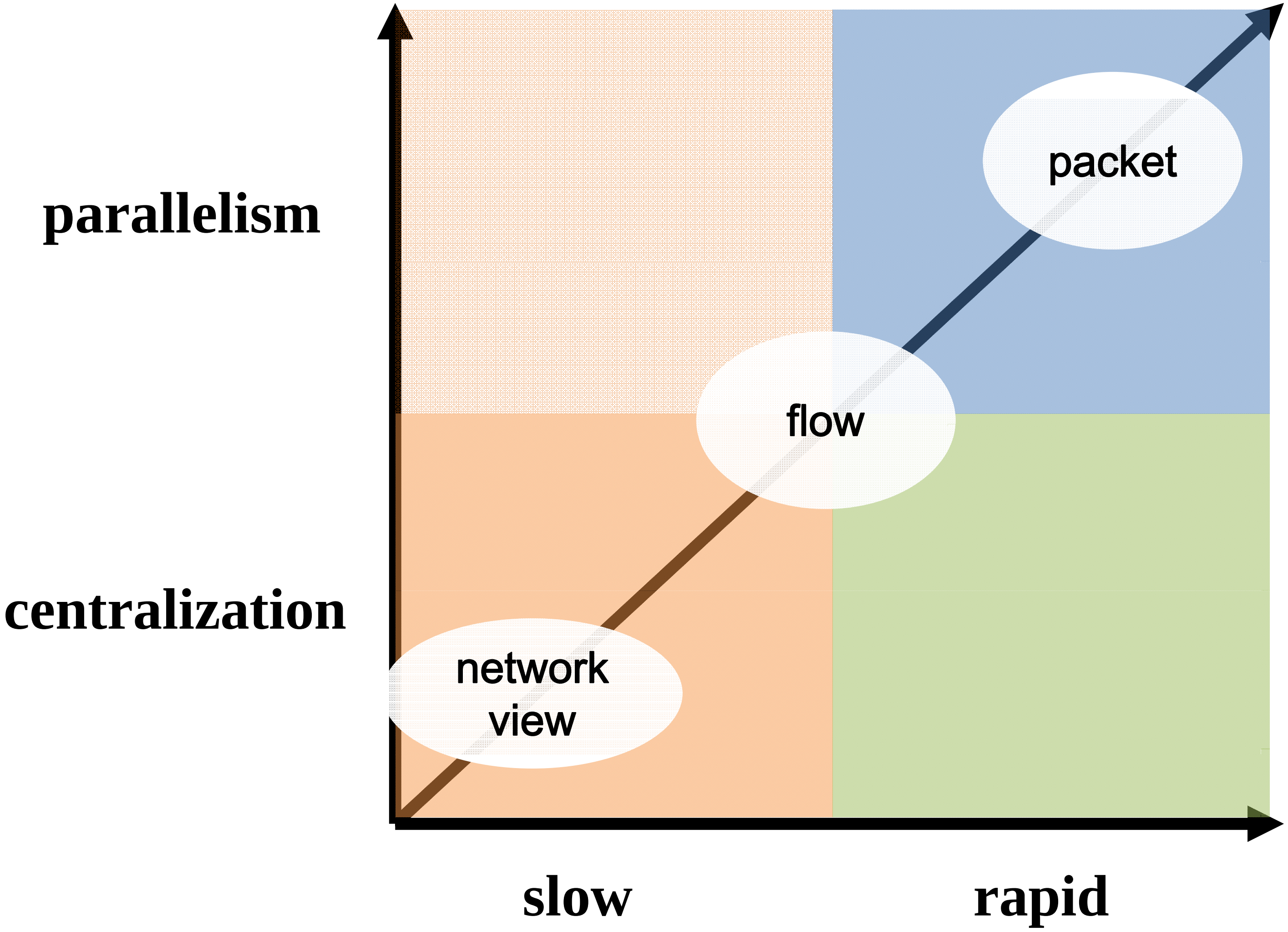
Header	Counter	Actions
Packet_hdr_info1	2	forward
Packet_hdr_info2	6	deny

Operation



Determine
Compute
Install…

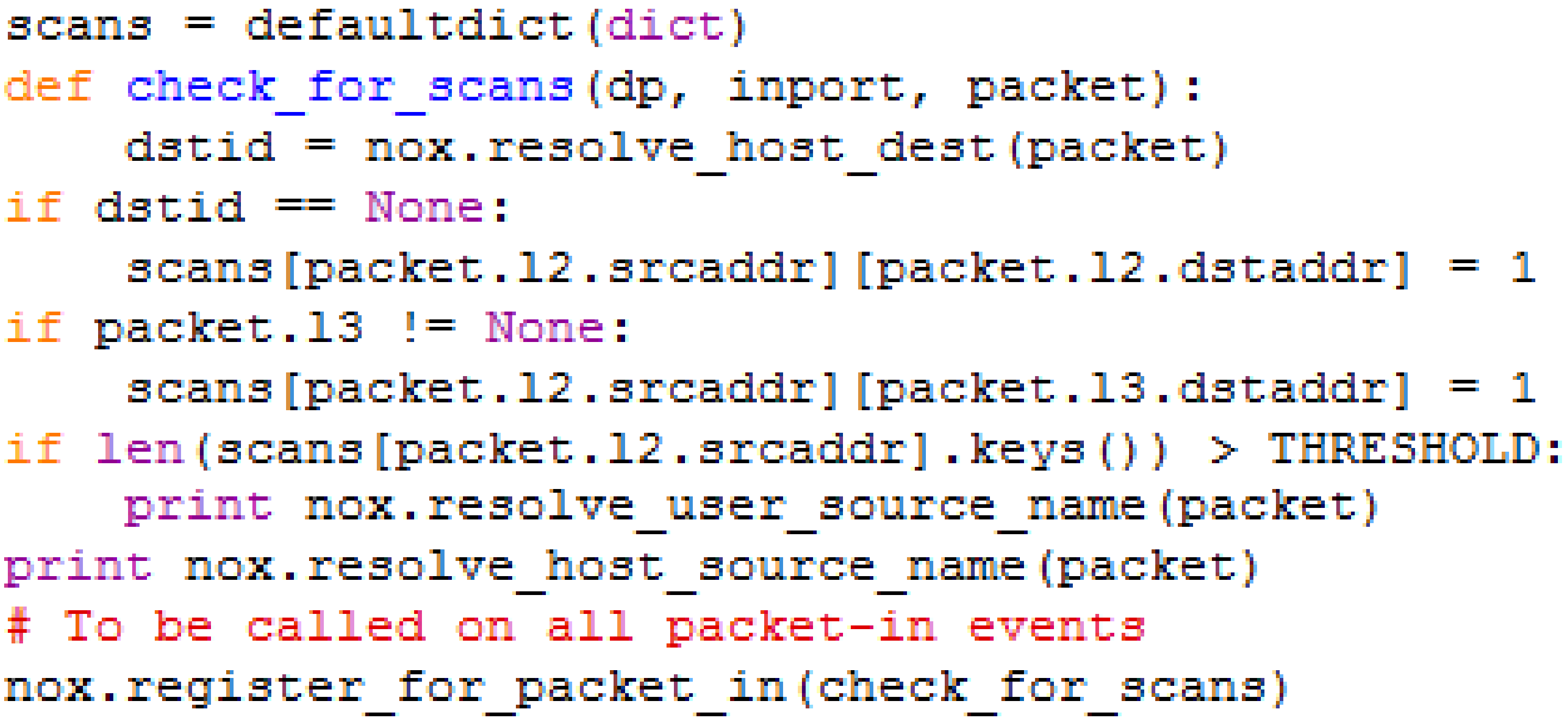
Scaling


parallelism
centralization
packet
flow
network view
slow
rapid

Programmatic Interface

- Events
- Namespace
- Libraries
  - Routing
  - packet classification
  - DNS
  - network filtering

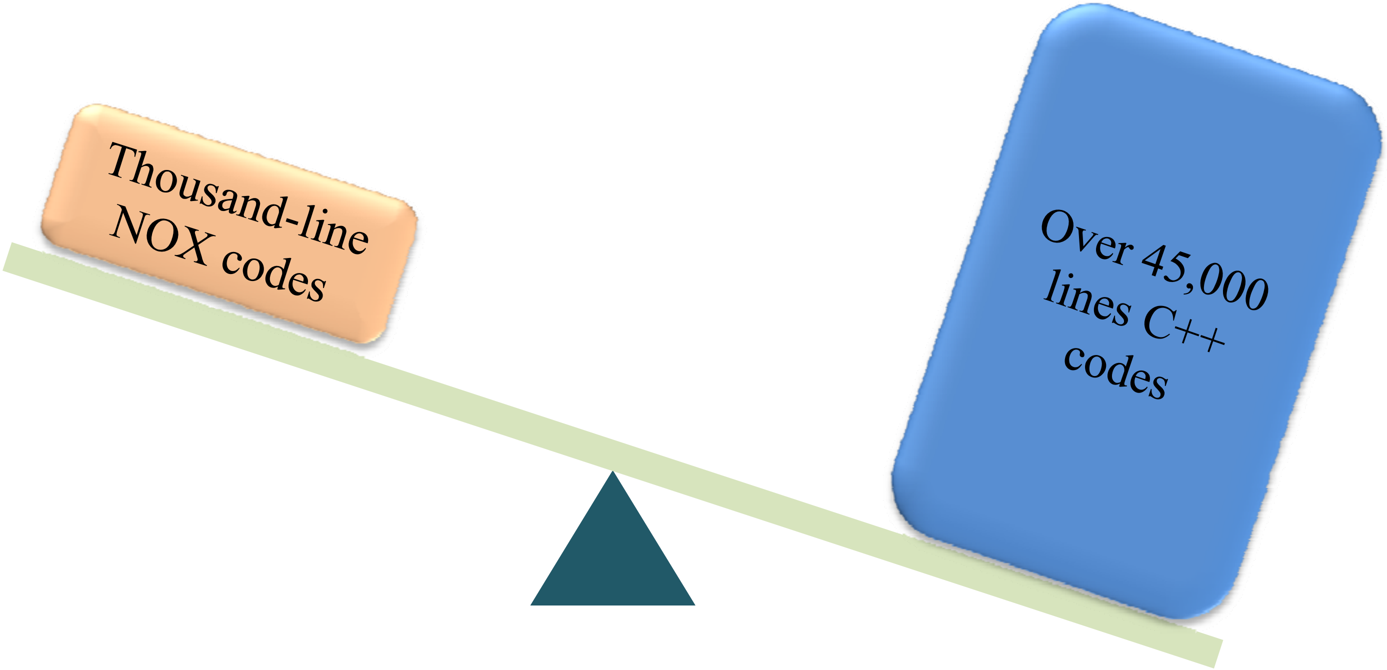
Detect Scanning hosts



```
scans = defaultdict(dict)
def check_for_scans(dp, inport, packet):
    dstid = nox.resolve_host_dest(packet)
if dstid == None:
    scans[packet.l2.srcaddr][packet.l2.dstaddr] = 1
if packet.l3 != None:
    scans[packet.l2.srcaddr][packet.l3.dstaddr] = 1
if len(scans[packet.l2.srcaddr].keys()) > THRESHOLD:
    print nox.resolve_user_source_name(packet)
print nox.resolve_host_source_name(packet)
# To be called on all packet-in events
nox.register_for_packet_in(check_for_scans)
```

Ehane

- One Network-wide Access-Control System



Summary

- High-level abstraction and easy programming
- Large-system scalability
- Not a replacement, but a cooperative framework of network management

References

- **NOX**
  - http://www.noxrepo.org
- **OpenFlow**
  - http://www.openflowswitch.org
- **4D**
  - M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe. Design and implementation of a routing control platform. In NSDI’ 05, 2005.
  - A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A Clean Slate 4D Approach to Network Control and Management. In ACM SIGCOMM Computer Communication Review, 2005.
- **Maestro**
  - Z. Cai, F. Dinu, J. Zheng, A. L. Cox, and T. S. E. Ng. Maestro: A Clean-Slate System for Orchestrating Network Control components. under submission, 2008

Q&A

?