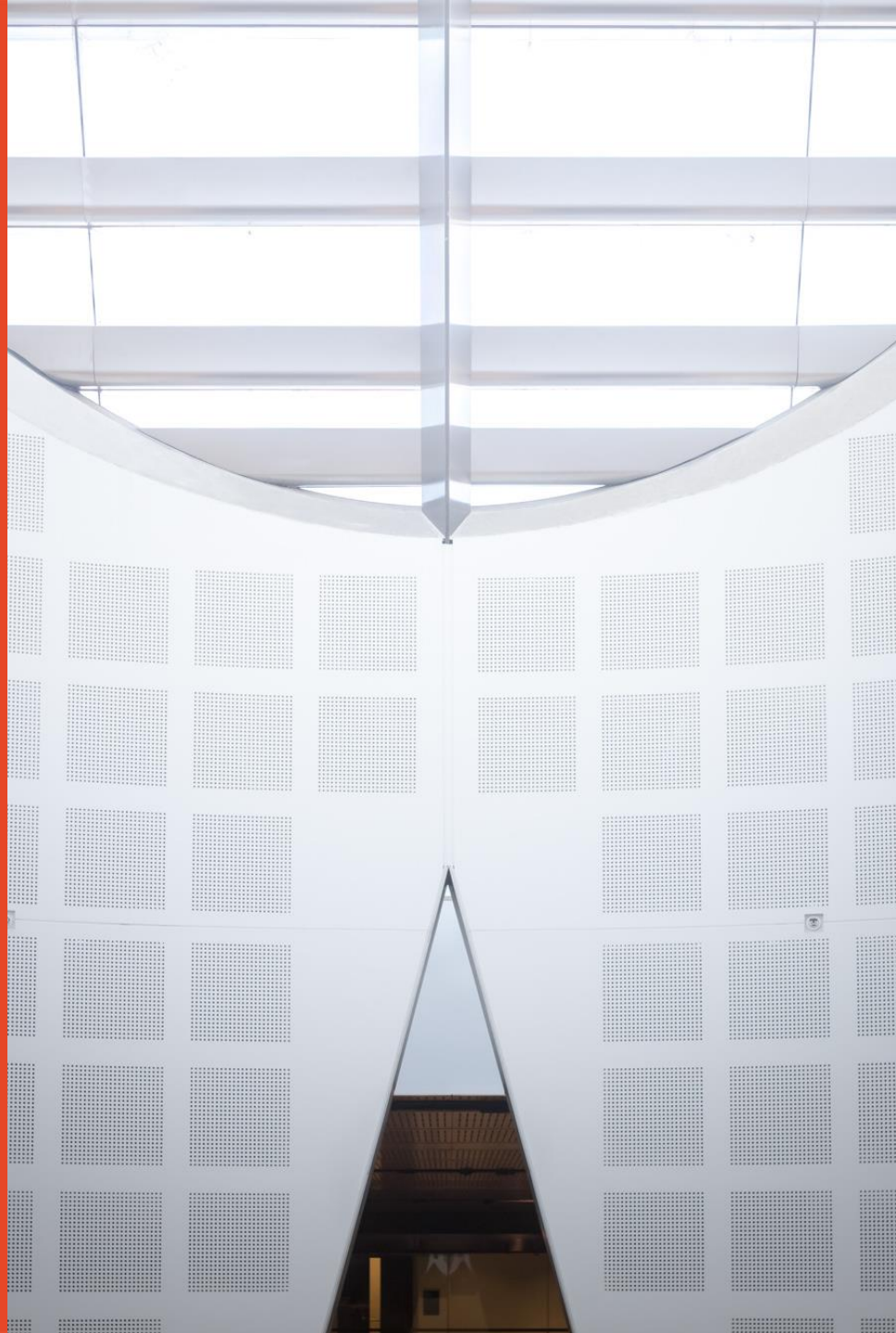
Topic 1: Introduction

Presented by
Dong YUAN

School of Electrical and Computer Engineering

dong.yuan@sydney.edu.au

THE UNIVERSITY OF SYDNEY



Contents

- About This Unit

- Introduction to Software Defined Networks

About Your Instructor

Dong Yuan

Associate Professor, School of Electrical and Computer Engineering

Contact Information:

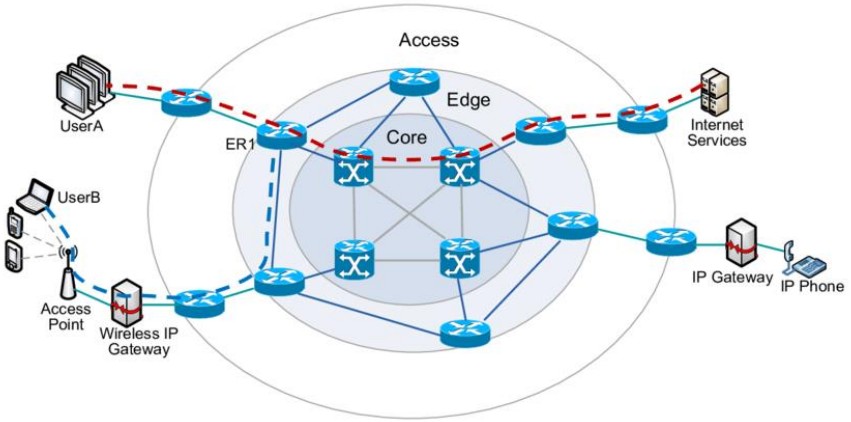
Email : dong.yuan@sydney.edu.au
Office : Room 323, PNR Building
Phone: 8627 2007

My research: Distributed systems, AI, Cloud/Edge computing, Internet of Things, etc.

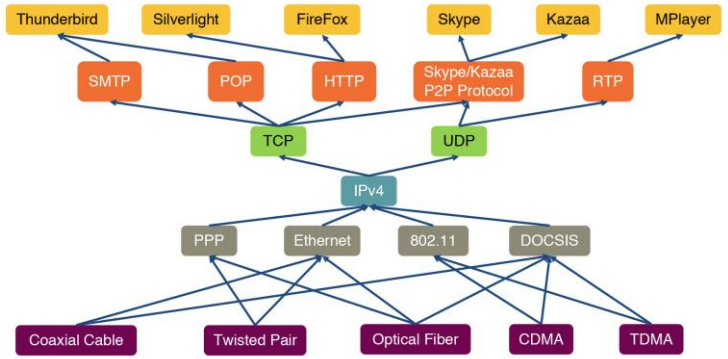
Tutors: Yanli Li yanli.li@sydney.edu.au

Internet Architecture

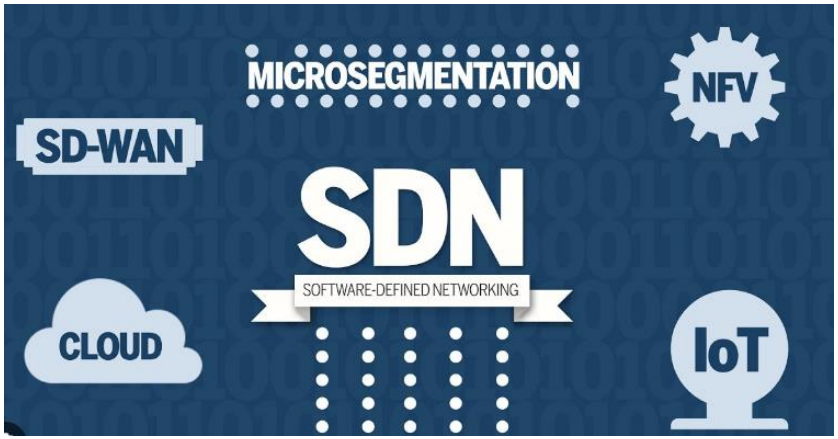
Hardware View



Protocol View



Today's Internet:



Advanced Objectives (My Expectation)

- How to design the interfaces
  - To the hardware (southbound)
  - API of the network operation system (northbound)
- Design of virtualization layer
- Design of the network operation system
- How to achieve isolation between different slices
- How to develop applications
  - Network programming language
- Security issues in SDN
- Etc.

Objectives (realistic ones)

- Teach/learn the fundamentals of software defined networking.
- Teach/learn the fundamentals of OpenFlow protocol.
- Teach/learn the most recent software defined networking applications.
- Practise with Mininet platform and development SDN project based on real world scenarios.
- Hands-on experience on SDN switches.

Topics covered

- Introduction to SDN
- Networking Technologies Basics
- OpenFlow and Mininet (Lab)
- Smart Switches
- Controllers Design
- Programmable Data Planes
- Network Virtualisation
- Network Function Virtualisation
- SDN applications
- SDN in Datacenters
- Advanced SDN programming
- SDN Security
- Advanced Topics (research)

Lab and Tutorial

- **Lab Sessions:** Practical exercises in lab about aspects of the unit and project.
  - Thursday 7pm-9pm (Weeks 2 - 13)

- **Software Tools/Services for Lab**
  - Virtual Box
  - Wireshark
  - Mininet with Linux
  - MiniEdit / Python

  - Opensource controllers
  - cloud networking services

Assessment

- Exams (50%) – Close Book
  - Weekly quiz (10%), 10 questions to be completed before each lab from week 3.
  - Final exam (40%), in exam period
- Project (50%) – group project individual marking
  - Project 1 (10%): Weeks 4-7
  - Project 2 (25%): Weeks 7-11
  - Project 3 (15%): Weeks 10-13

Course Prerequisites

- Familiarity with networking technologies

Contents

- About This Unit

- Introduction to Software Defined Networks

Internet History

- 1960s: packet-switching
- 1970s: Internetworking
  - Principles: autonomy, best effort model, stateless routers, decentralised control. (today's internet architecture)
- 1980s: TCP/IP deployment, DNS, growth of hosts
- 1990s: Commercialisation, the WWW
- 2000s to date: New apps, mobile networks

- Key Internet Problems
  - Security, DoS attacks
  - Hard to manage
  - Lack of availability & QoS guarantee

New Internet Architecture

- Cannot make changes to the fundamental technologies of Internet, e.g., IP, routing, etc.
  - High cost to adopt new tech
  - Risks of malfunctions
  - Commercial concerns: need to see the benefit

- Redesign the Internet Architecture
  - **Stanford's Clean Slate** Initiative
  - 100s of projects funded world-wide
  - WWW Conference and Journal

SDN History

- Evolution of key supporting network technologies

- Central network control (1980s)
  - Telephone network
  - AT&T network control point (NCP)
- Programmable networks (1990s)
  - Active networks
- Network virtualisation (1990s - 2000s)
  - XEN, VINI

Central network control (1980s)

- Network Control Point
  - AT&T Telephone network
  - Signaling at NCP
  - Control and Data sent over same channel
- Benefits
  - Network-wide observation
  - Direct connection and shorter circuit holding time
  - Easily determine busy/idle of a circuit
- Still used in routing 800 calls

Programmable Networks (1990s)

- Switches perform custom computation on packets
  - Track program running at each router
  - Custom processing functions run on routers
  - Packets are routed through programmable nodes
  - Program depends on the packet header
- Limitations
  - No clear applications (e.g., cloud datacenter)
  - Hardware is expensive. ASICs are widely used, whereas now FPGAs are very popular.
  - Programming language, end user as the programmer rather than the network operator
- OpenFlow is a more advanced technology

Network Virtualisation (1990s - 2000s)

- One or more logical network topologies on the same infrastructure
  - Virtual LANs, VMWare, Switchlets, VINI, etc.
- Benefits:
  - Multiple logical routers on a single platform
  - Resource isolation in CPU, memory, bandwidth, etc.
  - Customizable routing and forwarding software
  - Cheap hardware: General-purpose CPUs for control plane and FPGA for data plane.
- Key to SDN
  - Separate service from Infrastructure
  - Multiple controllers on a single switch
  - Logical network topologies

SDN and Network Function Virtualisation

- 2000s to date

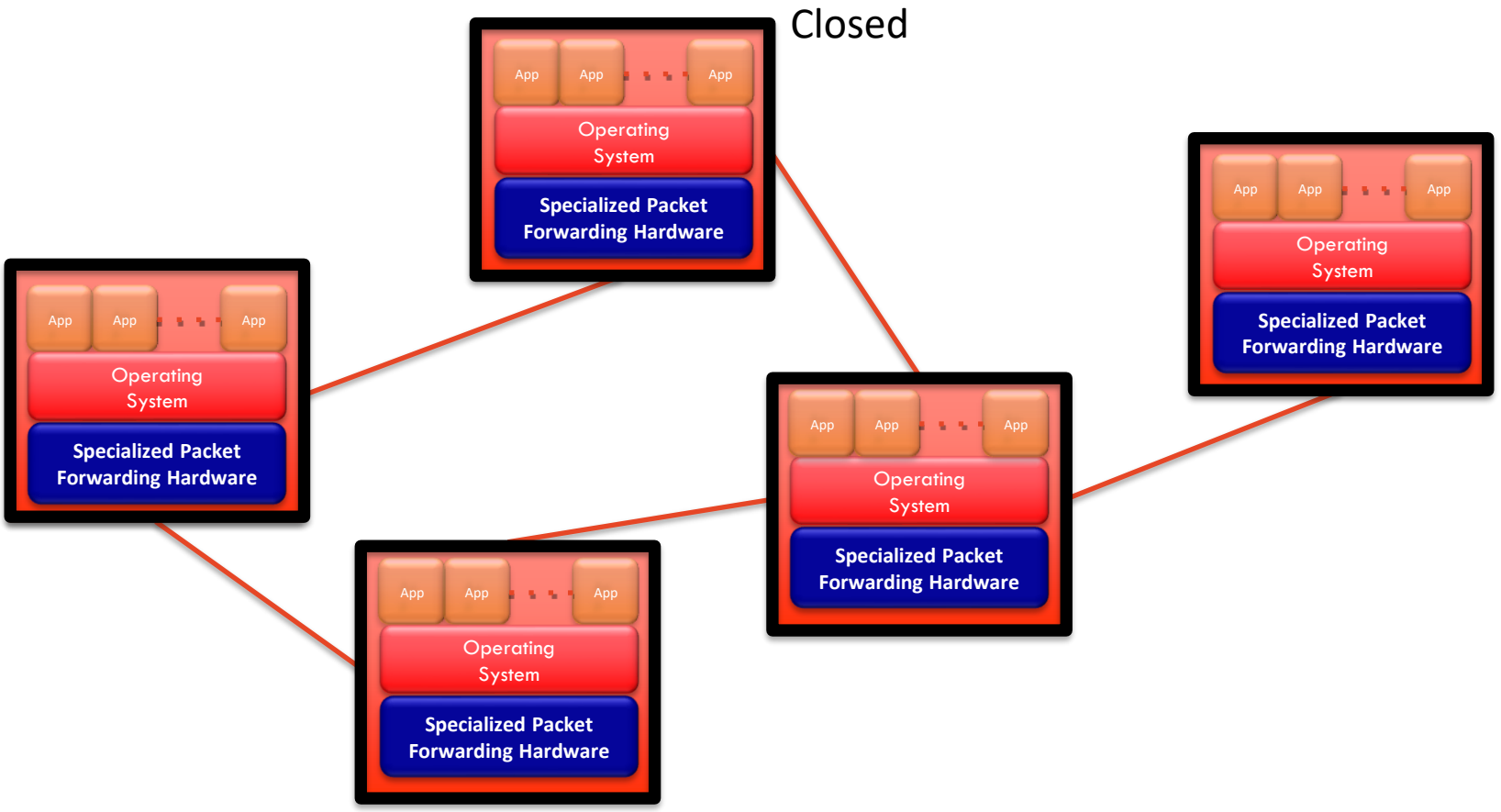
What is SDN?

- A new architecture that makes networks more programmable than in the past

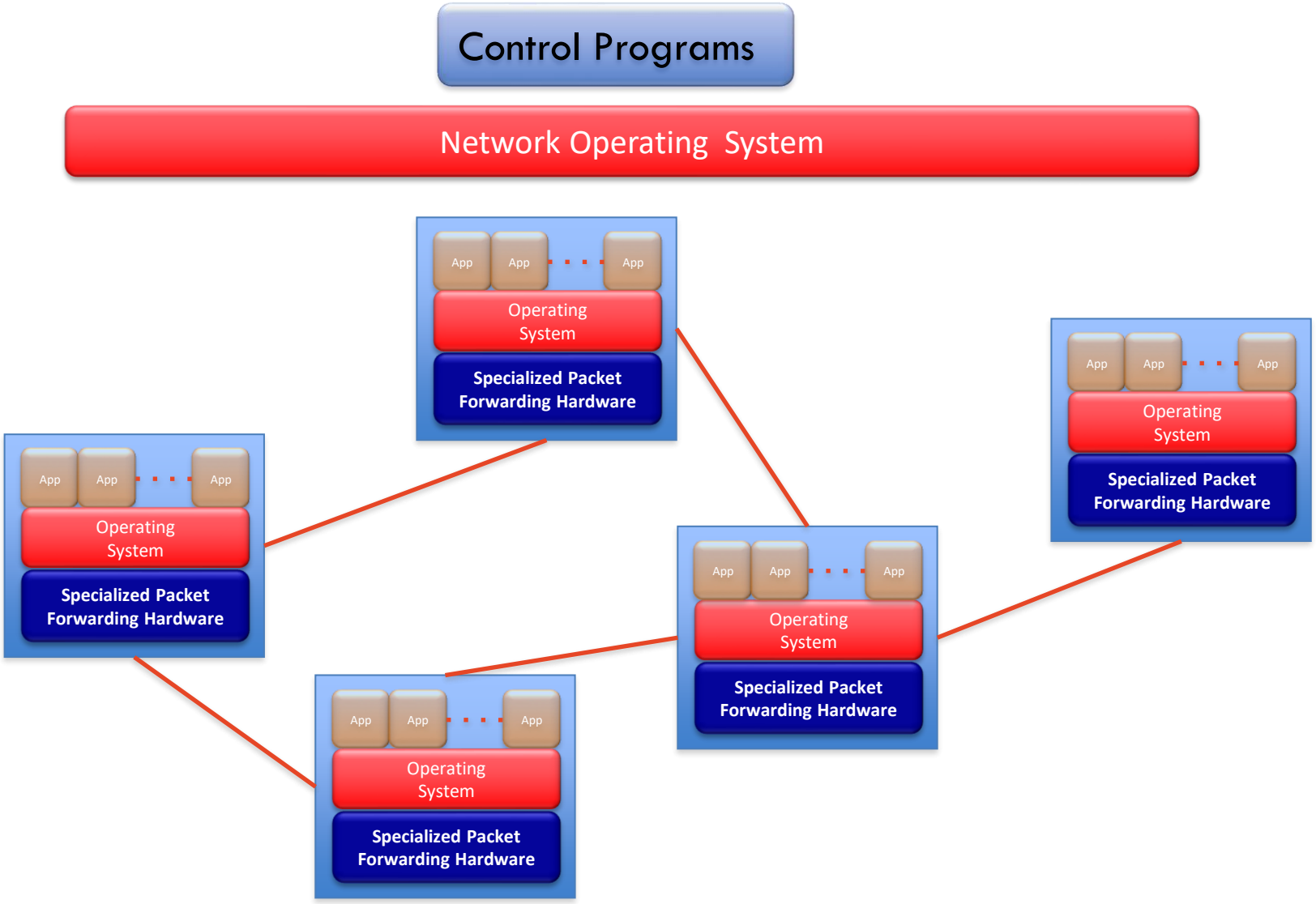
- Key principles:
  - Centralised control
  - **Open interfaces**
  - **Flow-based routing**

- **OpenFlow is the most widely adopted protocol.**
  - It is not SDN, but is one important SDN technology

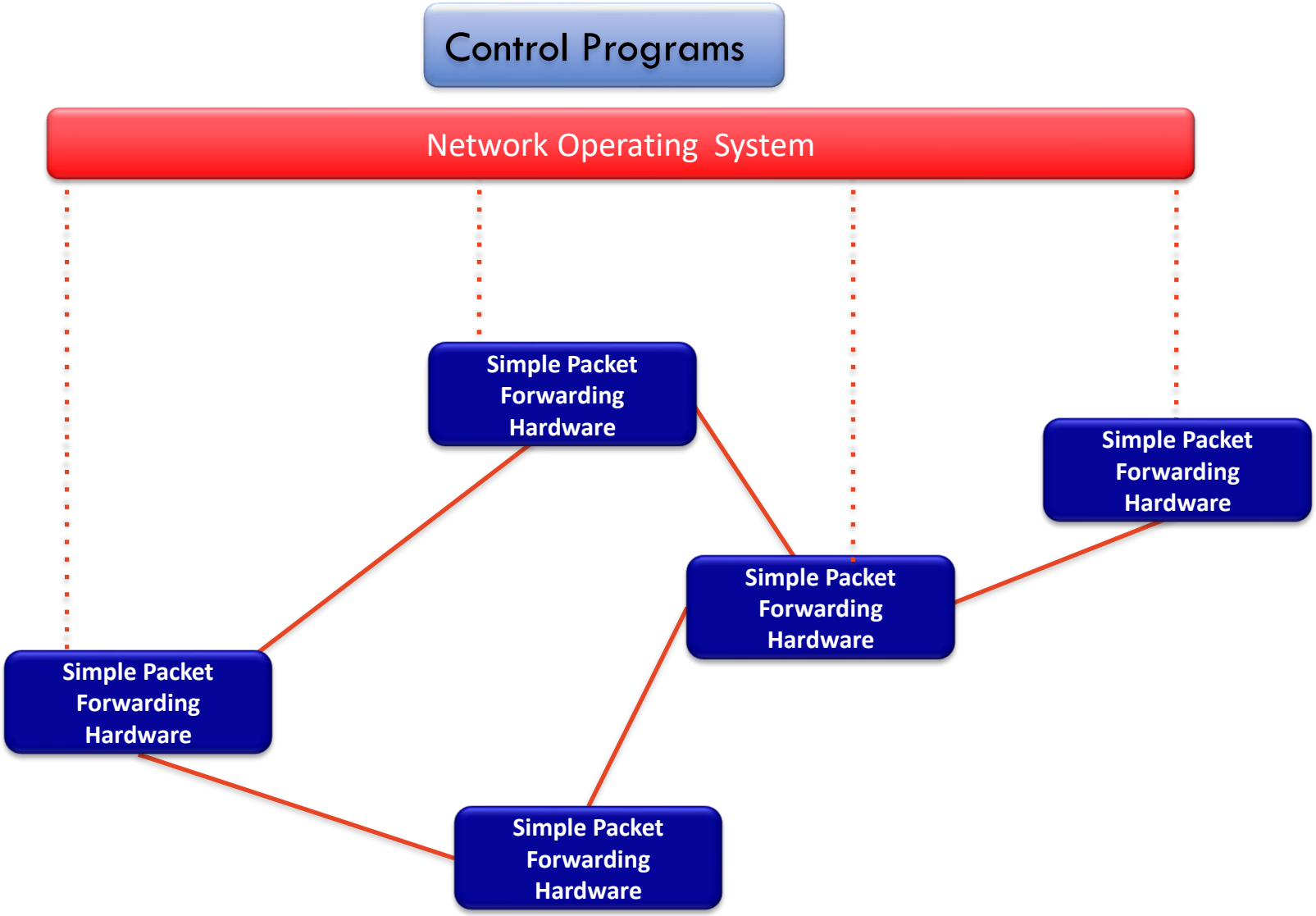
Idea: An OS for Networks



Idea: An OS for Networks



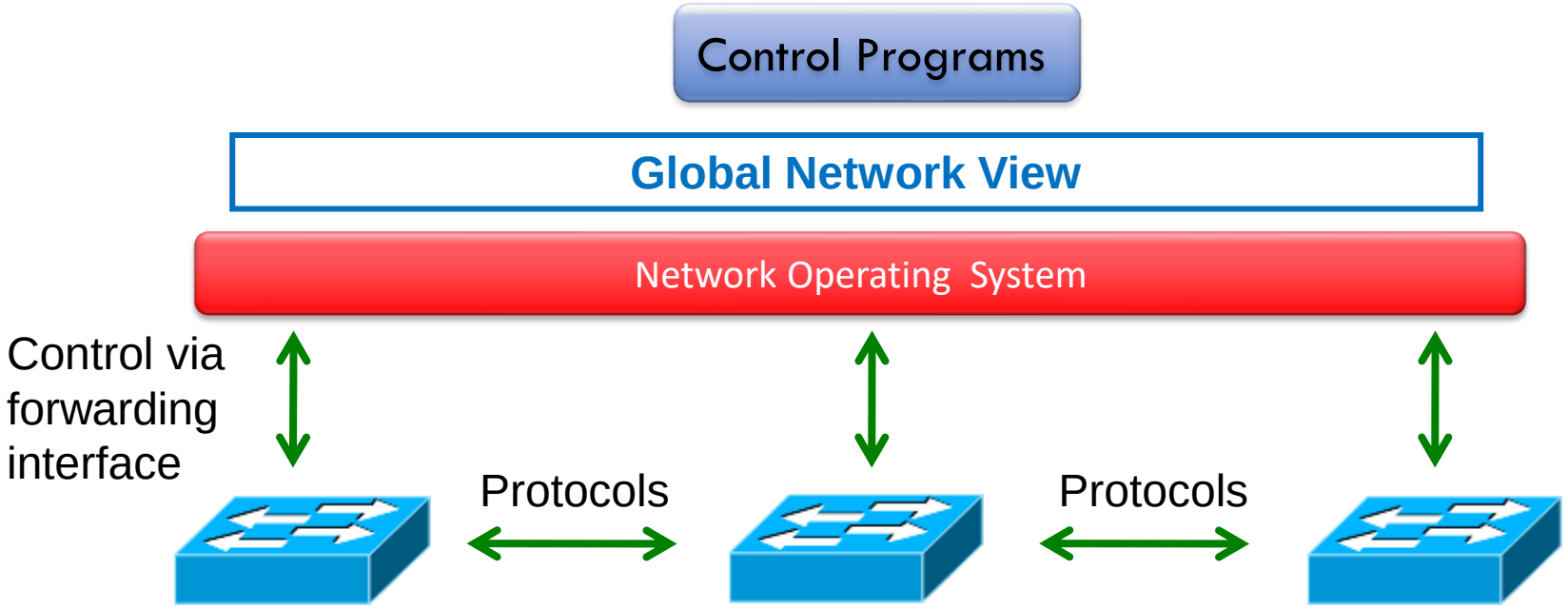
Idea: An OS for Networks



Idea: An OS for Networks

– **"NOX: Towards an Operating System for Networks"**

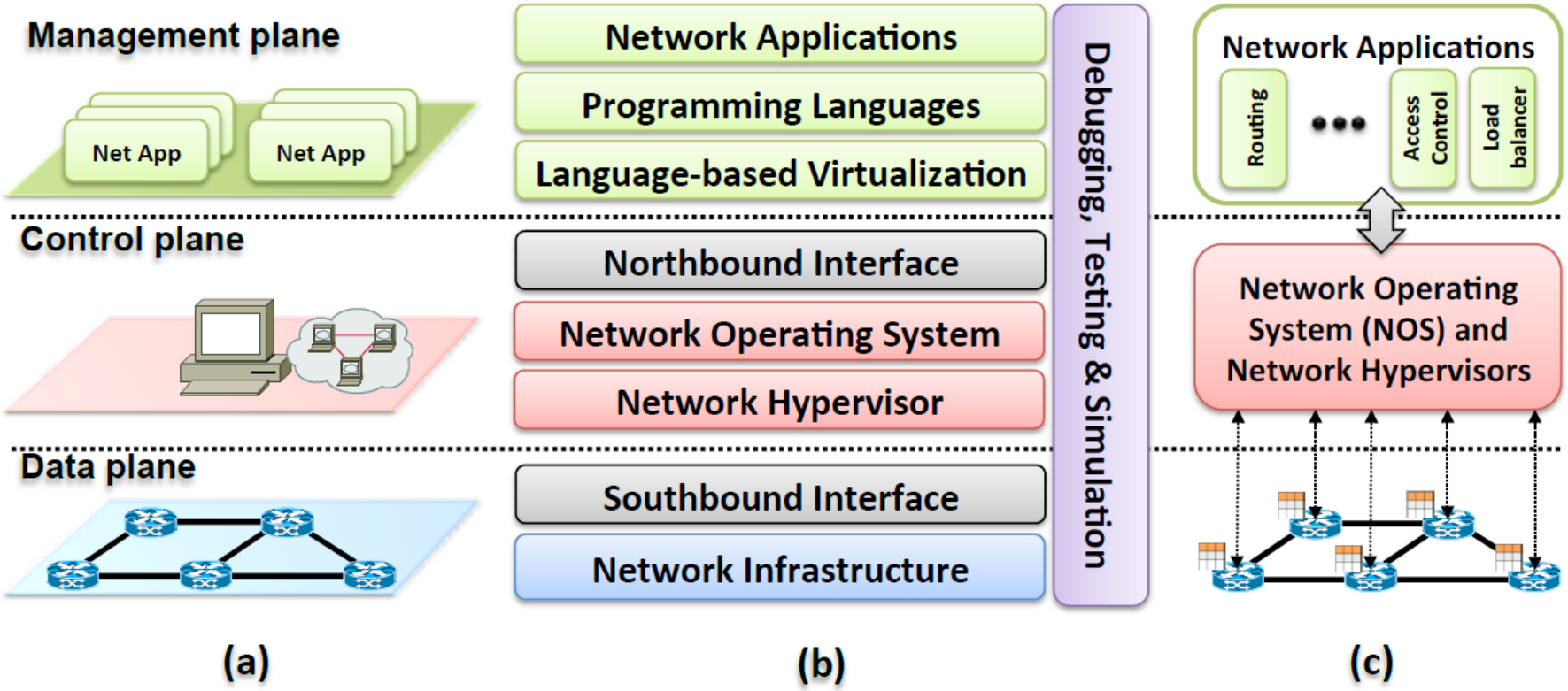
Software-Defined Networking (SDN)



SDN Basic Concept

- Separate Control plane and Data plane entities.
  - Network intelligence and state are logically centralized.
  - The underlying network infrastructure is abstracted from the applications.
- Execute or run Control plane software on general purpose hardware.
  - Decouple from specific networking hardware.
  - Use commodity servers and switches.
- Have programmable data planes.
  - Maintain, control and program data plane state from a central entity.
- An architecture to control not just a networking device but an entire network.

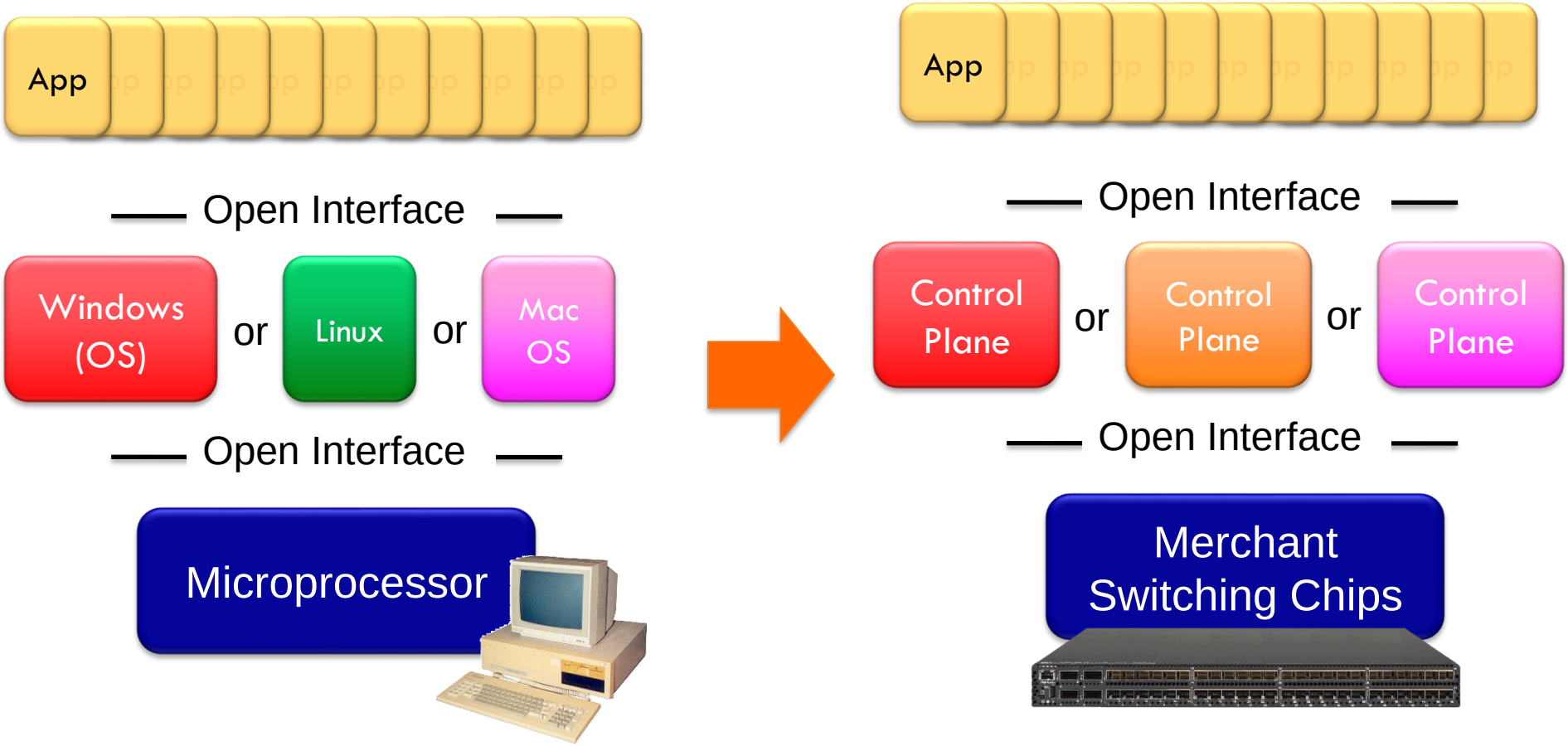
Software-Defined Networks



(a) planes, (b) layers, and (c) system design architecture

– Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE 103.1 (2014): 14-76.

Computer vs. SDN



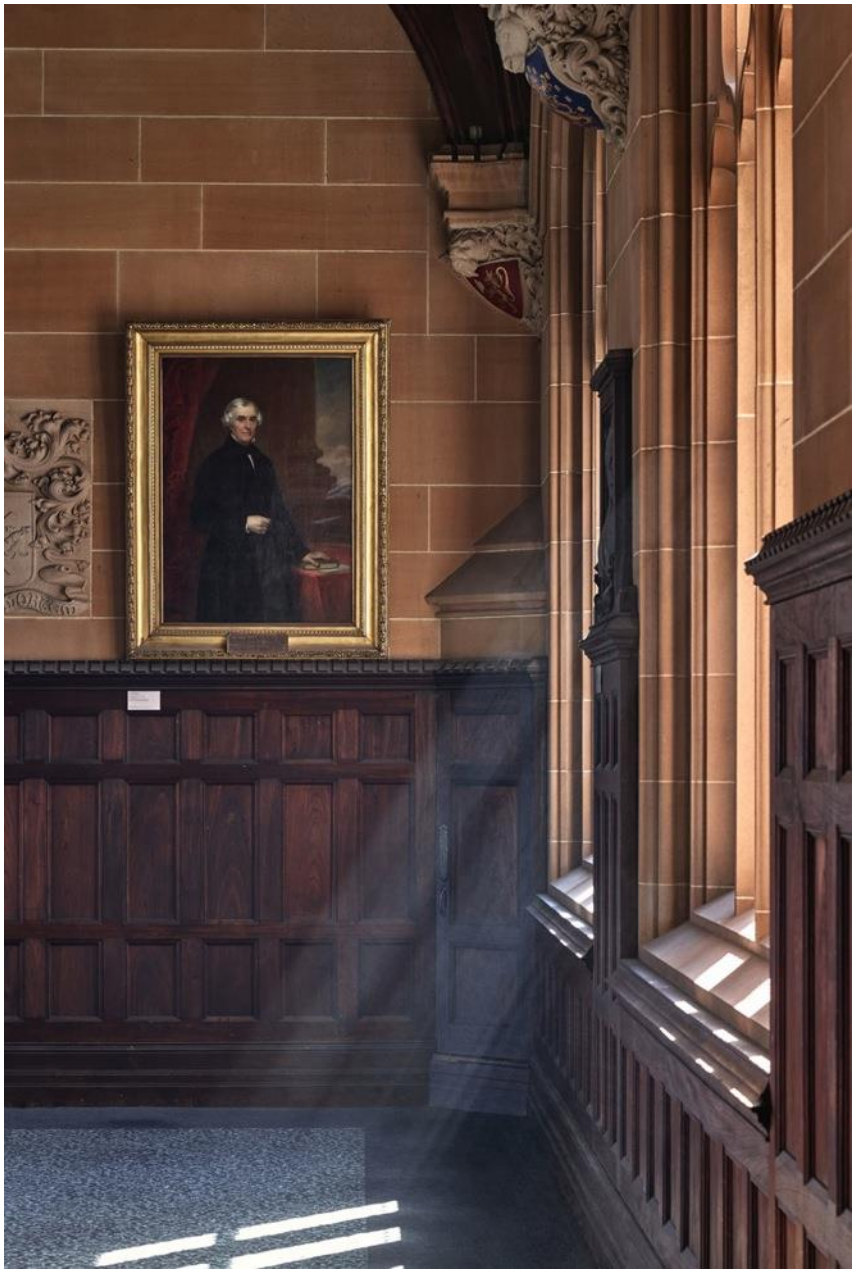
Open interfaces, Rapid innovation, Huge industry

SDN in Real World – Google's Story

- The industries were skeptical whether SDN was possible.
- Google had big problems:
  - **High financial cost** managing their datacenters: Hardware and software upgrade, over provisioning (fault tolerant), manage large backup traffic, time to manage individual switch, and a lot of men power to manage the infrastructure.
  - **Delay** caused by rebuilding connections after link failure.
    - Slow to rebuild the routing tables after link failure.
    - Difficult to predict what the new network may perform.
- Google went a head and implemented SDN.
  - Built their hardware and wrote their own software for their internal datacenters.
  - Surprised the industries when Google announced SDN was possible in production.
- How did they do it?
  - Read "*B4: Experience with a Globally-Deployed Software Defined WAN*", ACM Sigcomm 2013.

Thank you!

End



THE UNIVERSITY OF SYDNEY