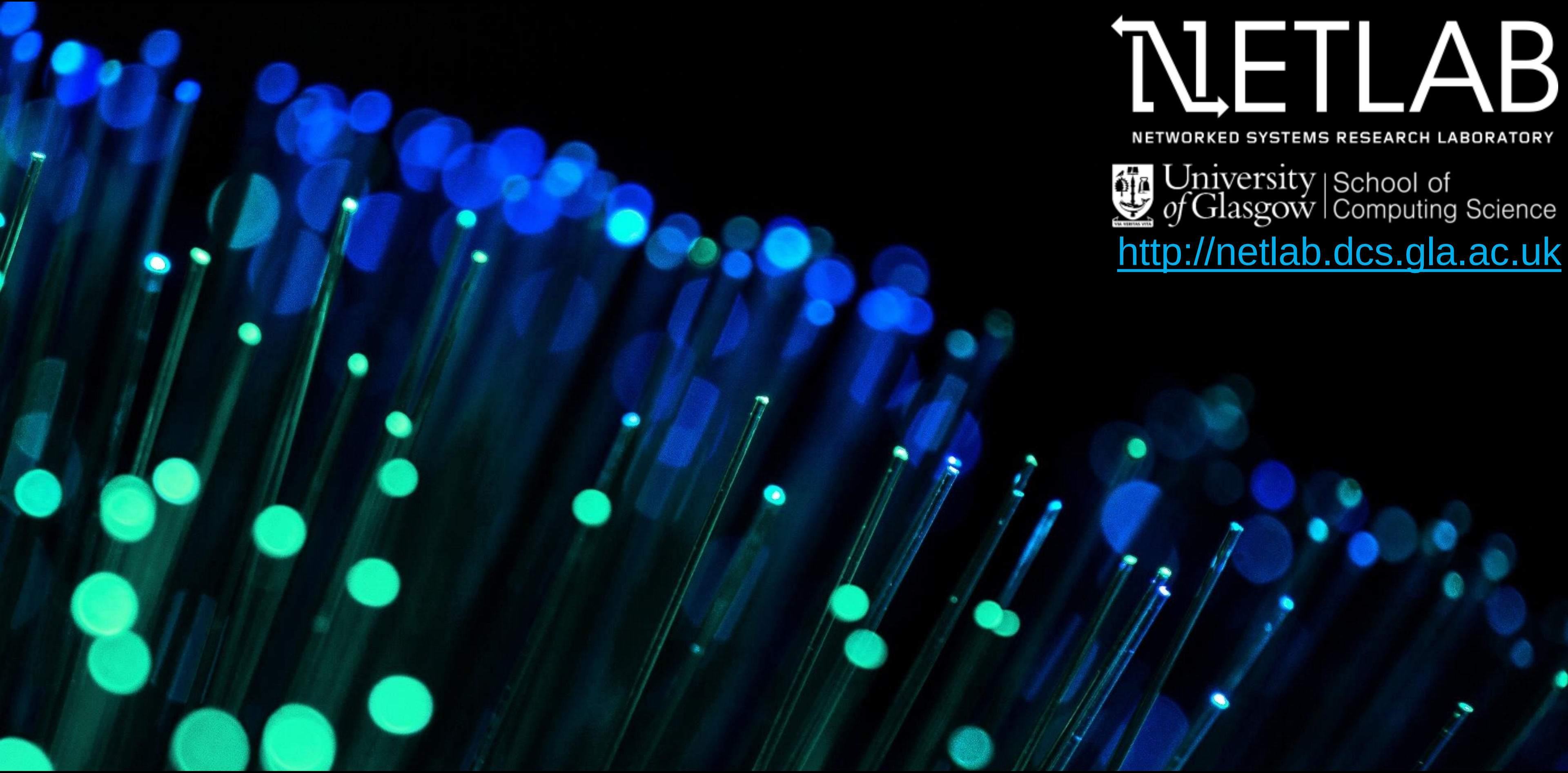


NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY
University of Glasgow | School of Computing Science
http://netlab.dcs.gla.ac.uk

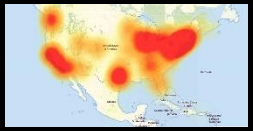
Infrastructure support for future resilient networked systems

Dr Dimitrios Pezaros | netlab | School of Computing Science

sicsa* The Scottish Informatics & Computer Science Alliance

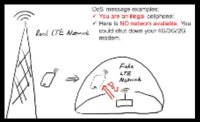
Cyber attacks and incidents intensify

- Dyn Cyberattack (2016)
  - Millions of Mirai-infected IoT devices overloaded Dyn's DNS servers bringing a big chunk of the Internet down (e.g., Spotify, Twitter, NYTimes, etc.)



- WannaCry ransomware attack (2017)
  - Encrypted data on MS Windows machines and propagated using a SMB protocol exploit; infected NHS, Telefónica, FedEx, etc.

- DDoS attack halts heating in Finland amidst winter (2016)
  - Attack disabled (overloaded) computers that were controlling heating in the buildings; went undetected for ca. two weeks



- Every LTE call, text, can be intercepted and blacked out (2016)
  - Base-station redirection functionality can be exploited by fake or rogue networks
- Major IT failures in mission-critical systems, e.g., airports (2017, 2014, 2013)

- 85% of cell towers offline in some Texas counties due to hurricane Harvey (2017)

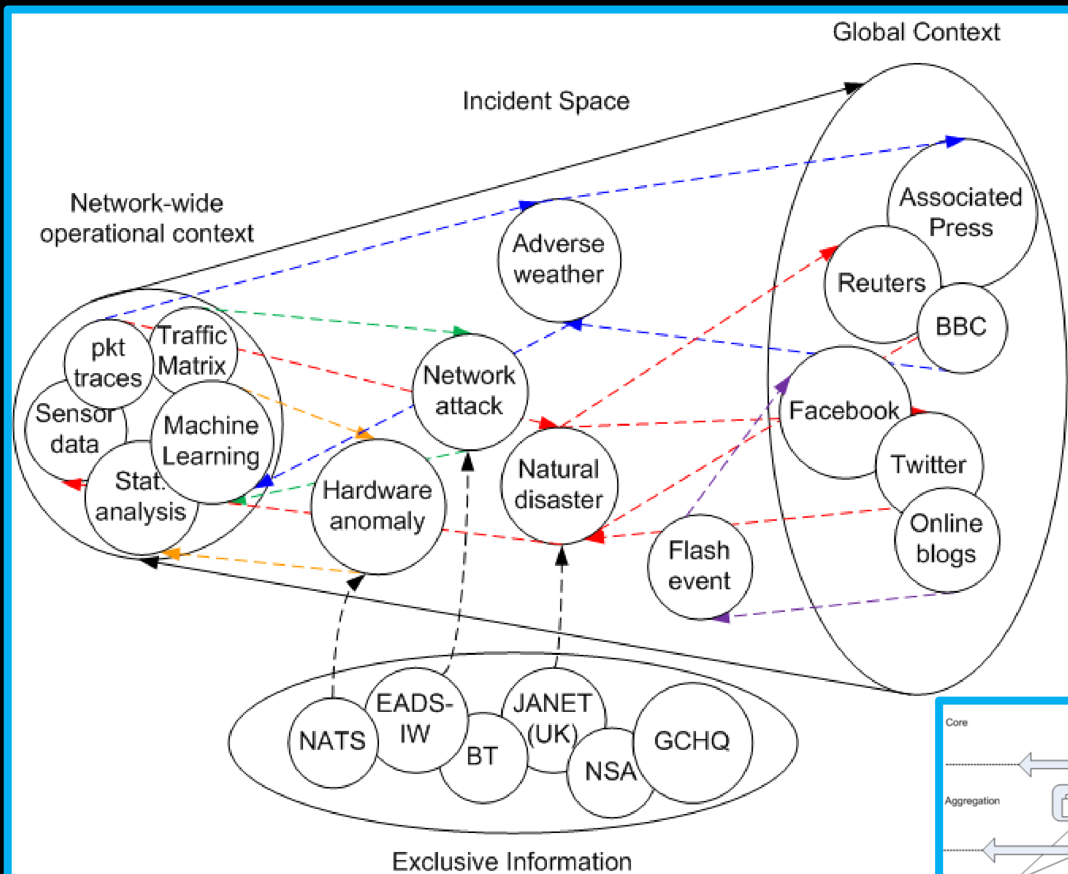
Resilience through Situation Awareness

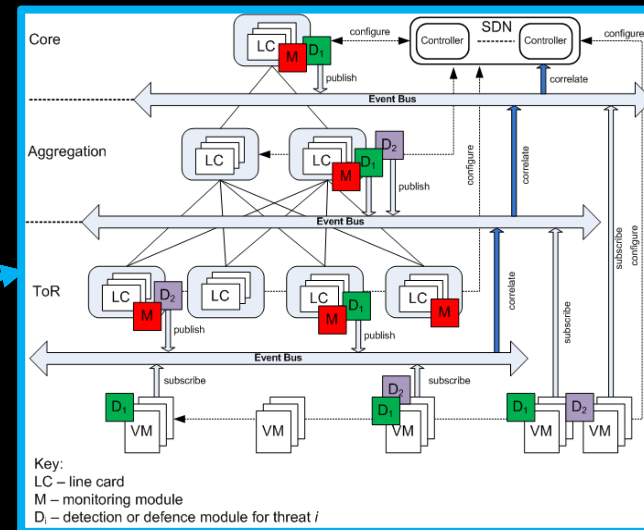
- Networked infrastructures become increasingly mission-critical:
  - Cloud DCs; ATC/ATM; SCADA; FTS
- Resilience and survivability paramount, but problematic
  - Network provisioning static and situation-agnostic
  - Anomaly/attack detection systems isolated; not integrated with network control algorithms
  - Bound to play cat-and-mouse with new threats and exploits
- Situation-awareness in terms of timely detection of and reaction to adversarial events
  - Self-* properties: learning, management, healing

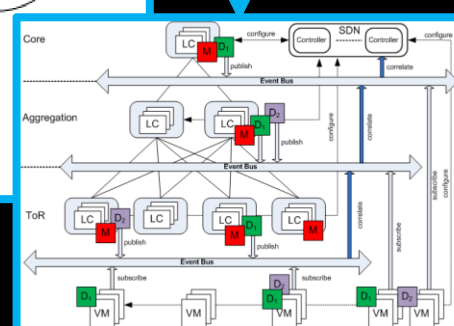
EPSRC Project: A Situation-Aware Information Infrastructure

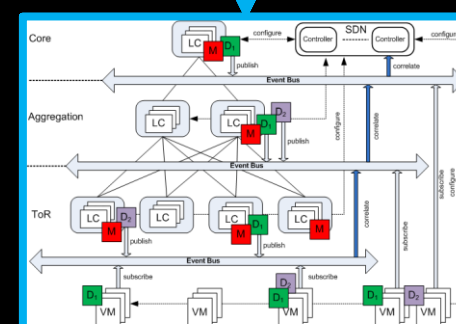
- Create an adaptive, situation-aware information infrastructure for future mission-critical networked environments
  - **Develop** an always-on, instrumentation and measurement infrastructure
  - **Develop** new statistical techniques to profile normal network-wide behaviour and detect adversarial incidents
    - ML, signal processing, information theory
  - **Develop** ways of modelling infrastructure-specific context from content analysis
    - Global feeds and operator explicit information
  - **Develop** network-wide situation-aware resilience mechanisms
    - Integrate situational awareness to the network control plane (e.g., routing)

Data and threat space; and architecture



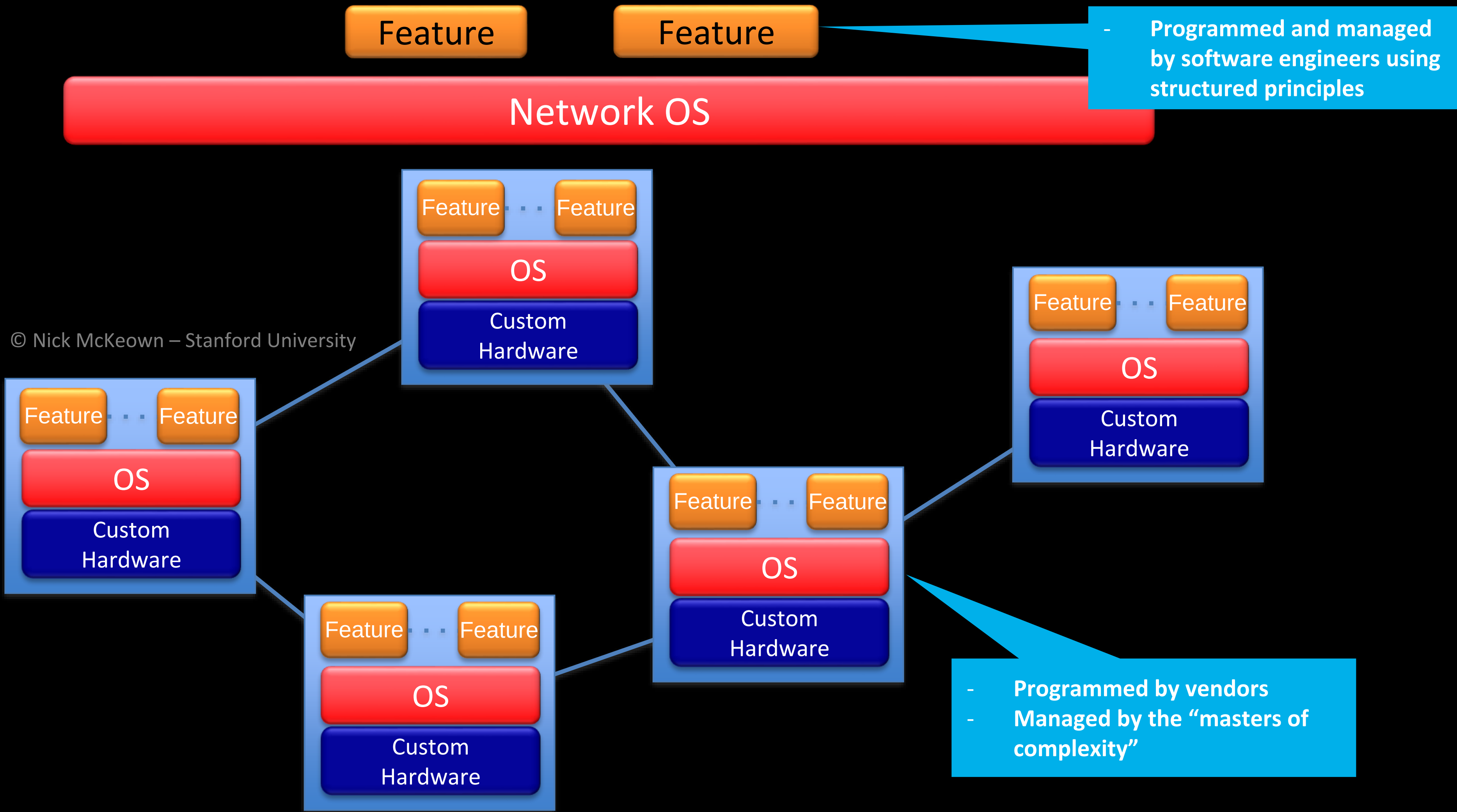




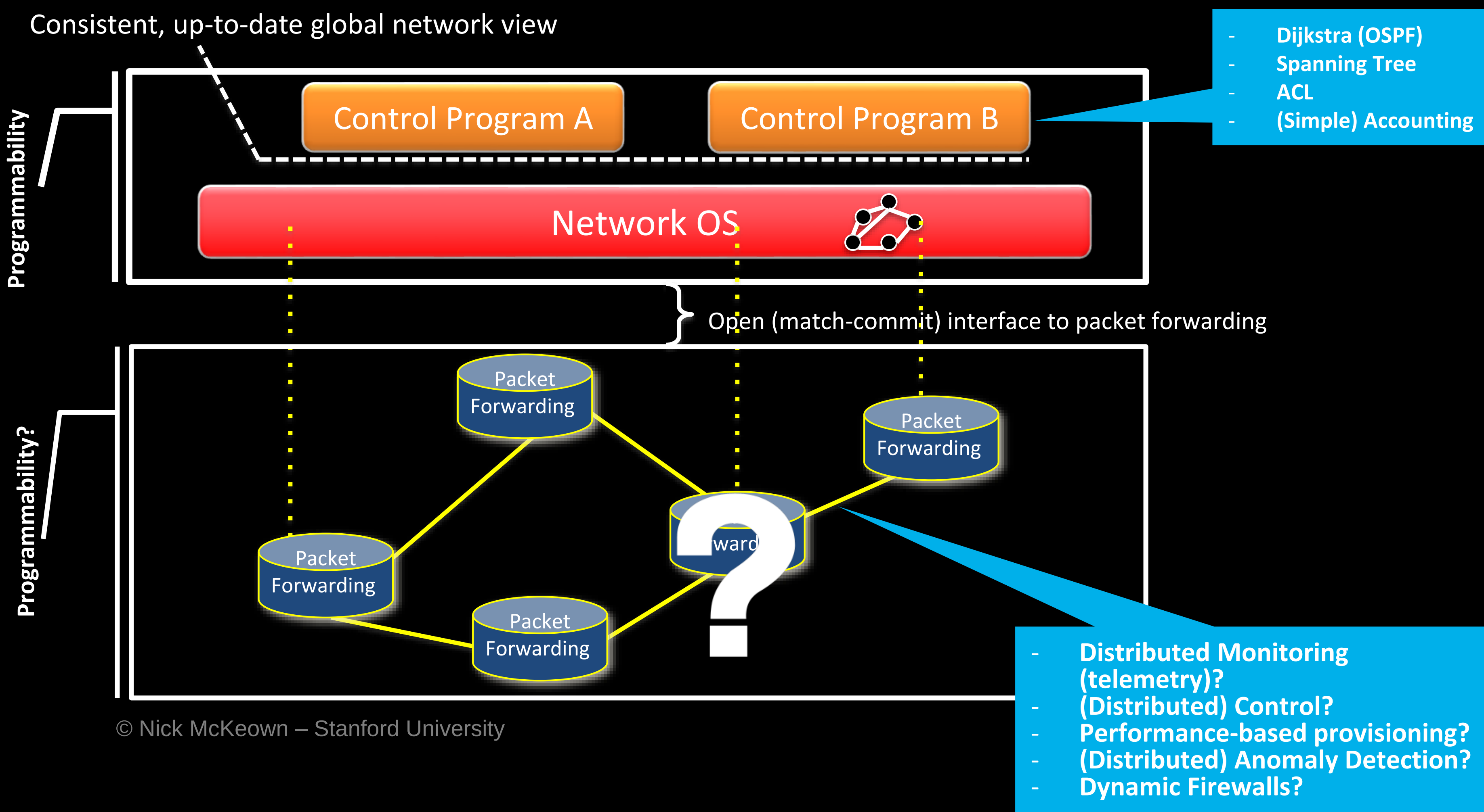


University of Glasgow

The network is changing – Software-Defined Networking (SDN)



SDN with OpenFlow – centralize network-wide decision-making



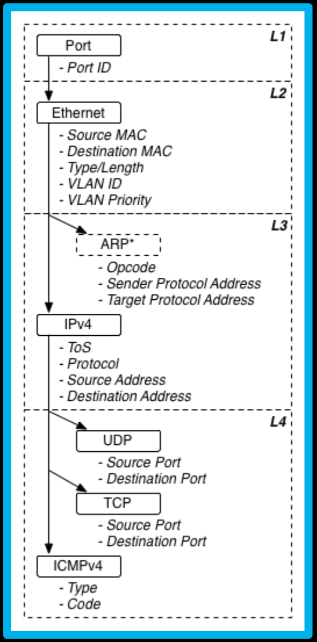
OpenFlow limitations – From delegating everything to delegating (almost) nothing

- OpenFlow allowed a lot of innovation in networking but far from perfect
  - Static, Incremental match field support
  - Tripled the memory required for a single flow entry

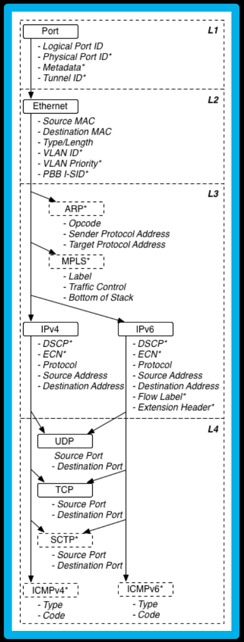
OF Version	Release date	Match fields	Depth	Size (bits)
<1.0	Mar 2008	10	10	248
1.0	Dec 2009	12	12	264
1.1	Feb 2011	15	15	320
1.2	Dec 2011	36	9–18	603
1.3	Jun 2012	40	9–22	701
1.4	Oct 2013	41	9–23	709
1.5	Dec 2014	44	10–26	773

  - Very limited functionality and purpose
  - Limited protocol support – 44 fields as OpenFlow 1.5
  - Limited matching - equality and bitmask only
  - No (line-rate) packet processing
  - Completely stateless

OpenFlow 1.0

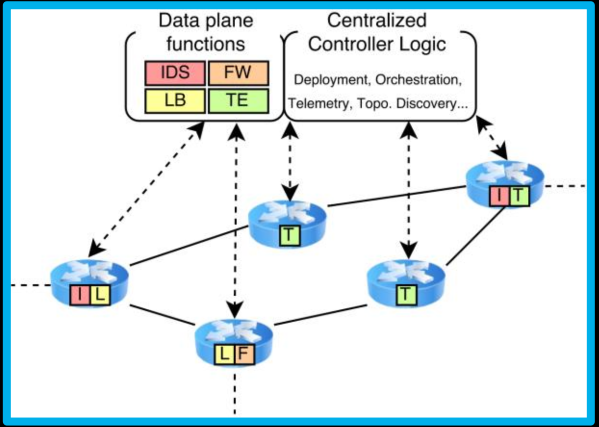


OpenFlow 1.5



Centralise knowledge, distribute (delegate) intelligence – BPFabric

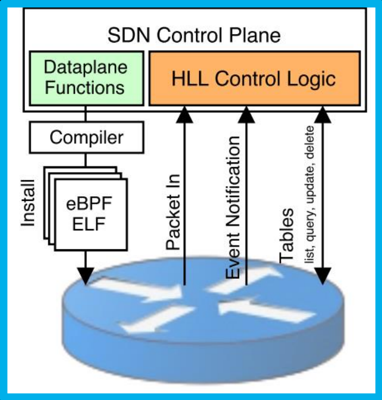
- Central controller can install data plane functions to the devices
- Programmable data plane through arbitrary packet matching and processing
  - Protocol-independent
  - Platform-independent
  - Language-independent
  - Stateful – tables for data storage and matching
  - Each data plane function is an acyclic control-flow graph



- Rapid introduction of new data plane functions
  - Routing and forwarding; middlebox-like functions currently not possible in OpenFlow (e.g., load-balancing, telemetry, debugging, security, QoS)

Use the eBPF instruction set to define per-switch packet processing pipelines

- cBPF/eBPF widely used instruction set(s) specifically defined for packet filtering
  - Pseudo-machine approach for protocol and platform independence (can then JIT/NPU/FPGA)
  - Close match to the instructions of a register machine interpretation fast and straight-forward



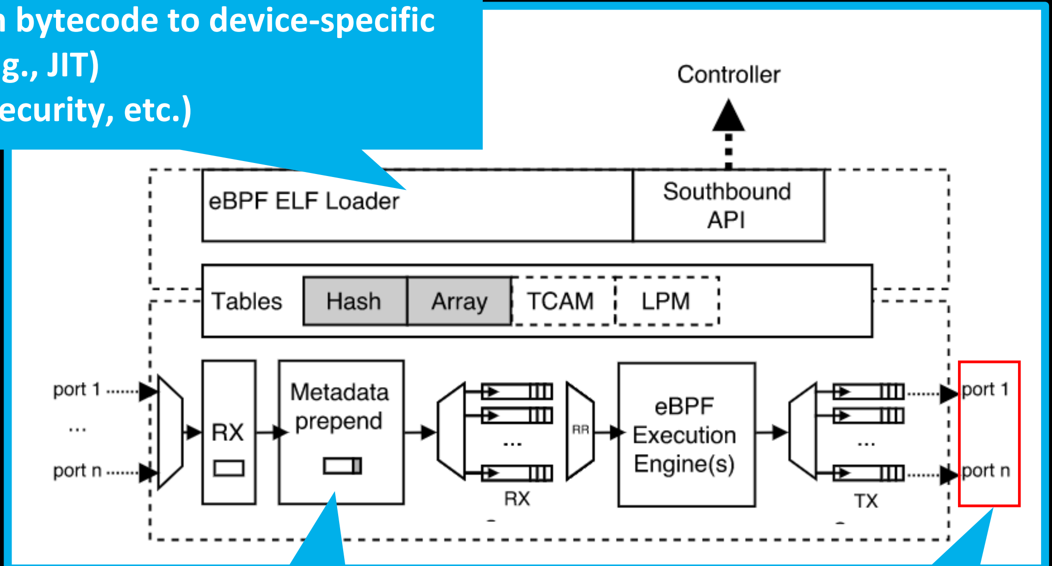
- Load (pkt header fields) and compare approach preventing backward jumps in the execution – deterministic execution time
  - BPF pipelines can be synthesised to aCFG(s) by combining the underlying parse, table, and conditional graphs.
- Controller defines network behaviour in HLL (C, P4, etc.)
  - Compile instruction set (eBPF); install function on switches; query/update/delete table entries
  - Switch executes ePBF for each packet (can raise events / ask controller / etc.)

University of Glasgow

Switch architecture

- Control Plane
  - eBPF Loader
  - BPFabric Agent
- Data Plane
  - Execute eBPF instructions for each packet received
  - Based on return code, forward to port, controller, flood or drop
- Lots more to say about the implementation(s)…
  - Controller interaction (protocol), performance, message types, example pipelines, etc.

1. Allocates the BPF tables required for the pipeline (described in the ELF metada)
2. Transform bytecode to device-specific format (e.g., JIT)
3. Verifier (security, etc.)



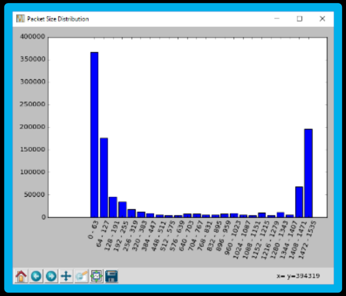
L1 metadata (timestamp, port, length, etc.) – info useful for much of the functionality

Physical and virtual ports (for flooding / controller / etc.)

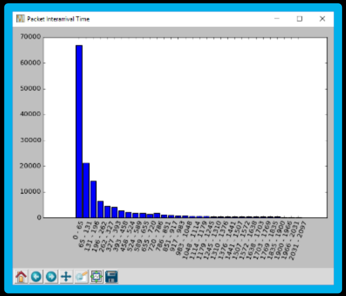
University of Glasgow

Example programs – network telemetry

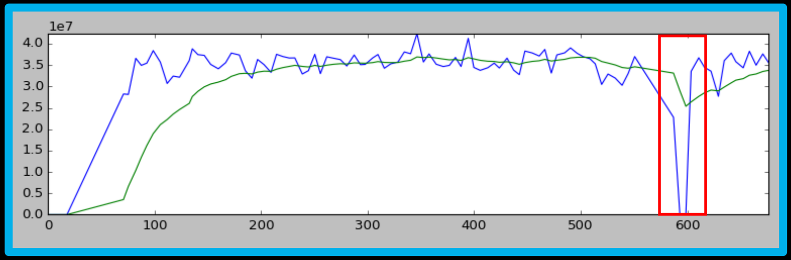
- Per-switch packet size distribution
  - Use an array type eBPF table to store histogram buckets
  - Controller can query the current state of the histogram (pull model)



- Per-switch packet interarrival time
  - Two tables to store the interarrival time histogram and time of last packet, respectively
  - Histogram pushed at periodically to the controller



- Lightweight anomaly detection
  - EWMA calculation of the incoming traffic volume for every port of a switch; maintained in an array map holding one entry per port
  - If computer value not within expected bounds, raise notification to the controller



SDN, NFV, and adaptive resource provisioning

- There have been significant advances in technology that can help put everything together and design resilient systems
  - Network Function Virtualisation (NFV)
    - Middlebox functionality is software-ised and can be flexibly deployed anywhere
    - Even on lightweight devices (e.g., IoT gateways, Raspberry Pi's, etc.)
  - Converged server/network resource management
    - SDN can be the central nervous system of the infrastructure
    - Monitor and control services and users
  - Resilience as a Service
    - Where, in the network, do we deploy anomaly detection and mitigation modules?
    - Who to protect, e.g., user vs. infrastructure

NETLAB

NETWORKED SYSTEMS RESEARCH LABORATORY

University of Glasgow | School of Computing Science

http://netlab.dcs.gla.ac.uk

Thank You

Questions?