NETLAB

NETWORKED SYSTEMS RESEARCH LABORATORY

University of Glasgow | School of Computing Science

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

Next Generation Networked Systems:

Virtualised – Programmable – Adaptive – Intelligent – Resilient

Dr Chris Anagnostopoulos

Virtualised & Programmable
Networked Systems

establish

Adaptive, Intelligent & Resilient
Next Generation Networked Systems

System & Algorithmic Perspective

Virtualised Networked Systems

[V]: The Environment Today

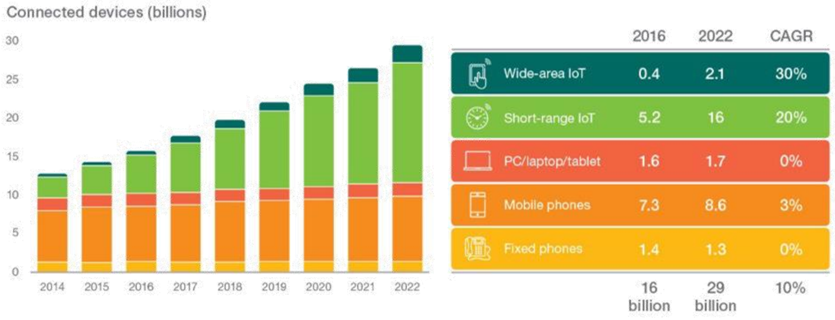
Customer Expectations put emphasis on (value-add) service provisioning:

- Low latency, high throughput (services)
- Development of new applications, e.g., Tactile Internet (H2M/M2M), personalized firewalls, VR/AR applications, HQ Video encoders …

Subject to: utilisation / devices increase dramatically…

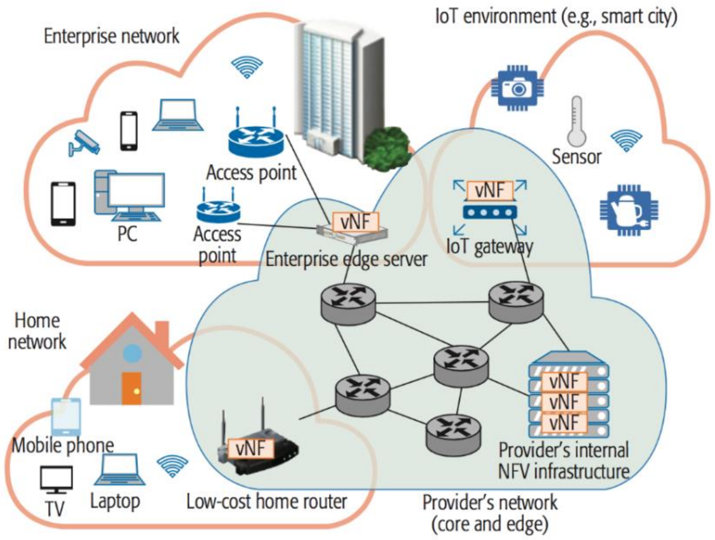
Key to Success: Fast Service Creation, Management & Intelligence

- **Goal: Decouple network functionality from physical locations for faster and flexible network service provisioning.**
- **How**: Lightweight Network Function Virtualisation (NFV)
- **Which**: vNFs, e.g., firewalls, caches, intrusion detectors, analytics
  - Support heterogeneous, resource-constrained networking environments, e.g., UxVs, WSNs, … (from Cloud to Edge);
  - Provide roaming & intelligent, optimal placement of vNFs;
  - Achieve Self-* (healing, learning, protection) properties based on in-network processing & data plane programmability



	2016	2022	CAGR
Wide-area IoT	0.4	2.1	30%
Short-range IoT	5.2	16	20%
PC/laptop/tablet	1.6	1.7	0%
Mobile phones	7.3	8.6	3%
Fixed phones	1.4	1.3	0%
	16 billion	29 billion	10%

Source: Ericsson IoT forecast



'Edge NFV Architecture'

[V]: Glasgow virtual Network Functions (GNF)

Goal: Bringing NFV to Network Edge

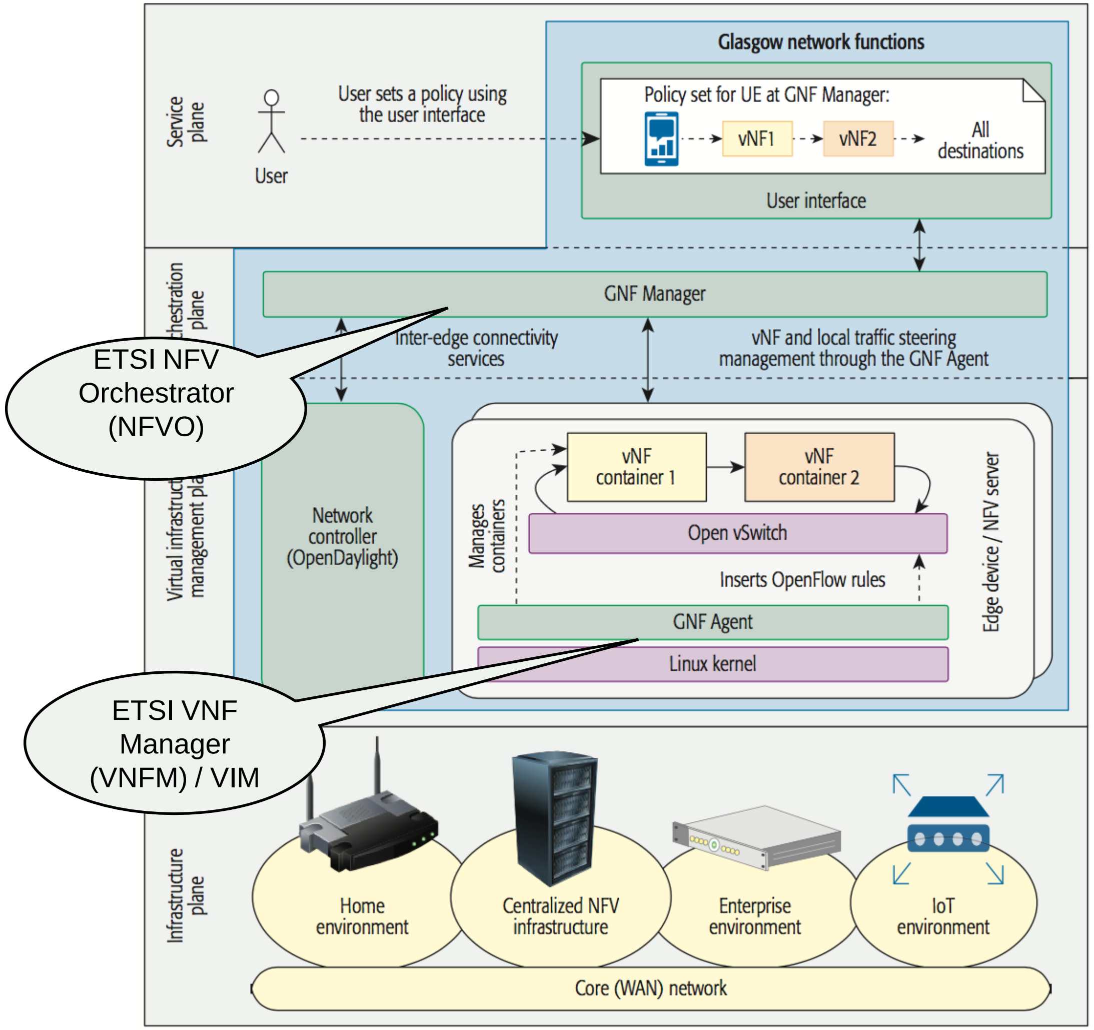
Service Plane: High-level administration access

- GUI representation of all connected devices; create vNF chains; assign vNFs to Edge devices (e.g., UxVs)

Orchestration (Knowledge) Plane: network-wide knowledge of vNF locations, usage statistics, etc.

- GNF Manager: REST APIs to start/stop/migrate vNFs
- vNFs Orchestration & Optimal Placement as **close to user** as possible s.t. resource constraints & triggered by e.g., user mobility, change in device utilization (energy budget in UxVs), etc.

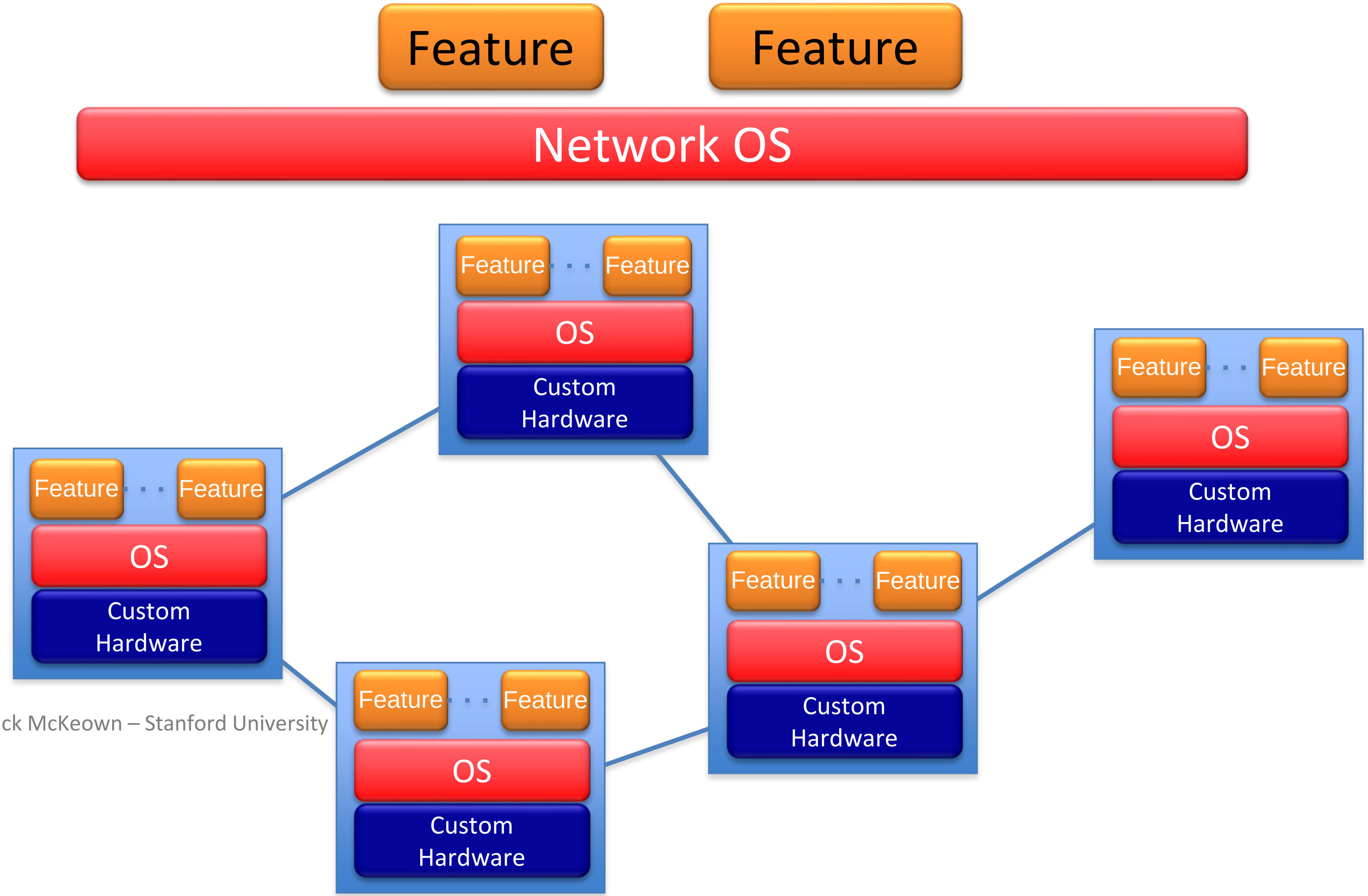
Virtual Infrastructure Management Plane: handles network connectivity between Edge Devices and the Central NFV Infrastructure, and the management of vNFs…



Service plane
User
User sets a policy using the user interface
Glasgow network functions
Policy set for UE at GNF Manager:
vNF1
vNF2
All destinations
User interface
Orchestration plane
GNF Manager
Inter-edge connectivity services
vNF and local traffic steering management through the GNF Agent
ETSI NFV Orchestrator (NFVO)
Virtual infrastructure management plane
Network controller (OpenDaylight)
Manages containers
vNF container 1
vNF container 2
Open vSwitch
Inserts OpenFlow rules
GNF Agent
Linux kernel
Edge device / NFV server
ETSI VNF Manager (VNFM) / VIM
Infrastructure plane
Home environment
Centralized NFV infrastructure
Enterprise environment
IoT environment
Core (WAN) network

Programmable (Software-Defined) Networked Systems

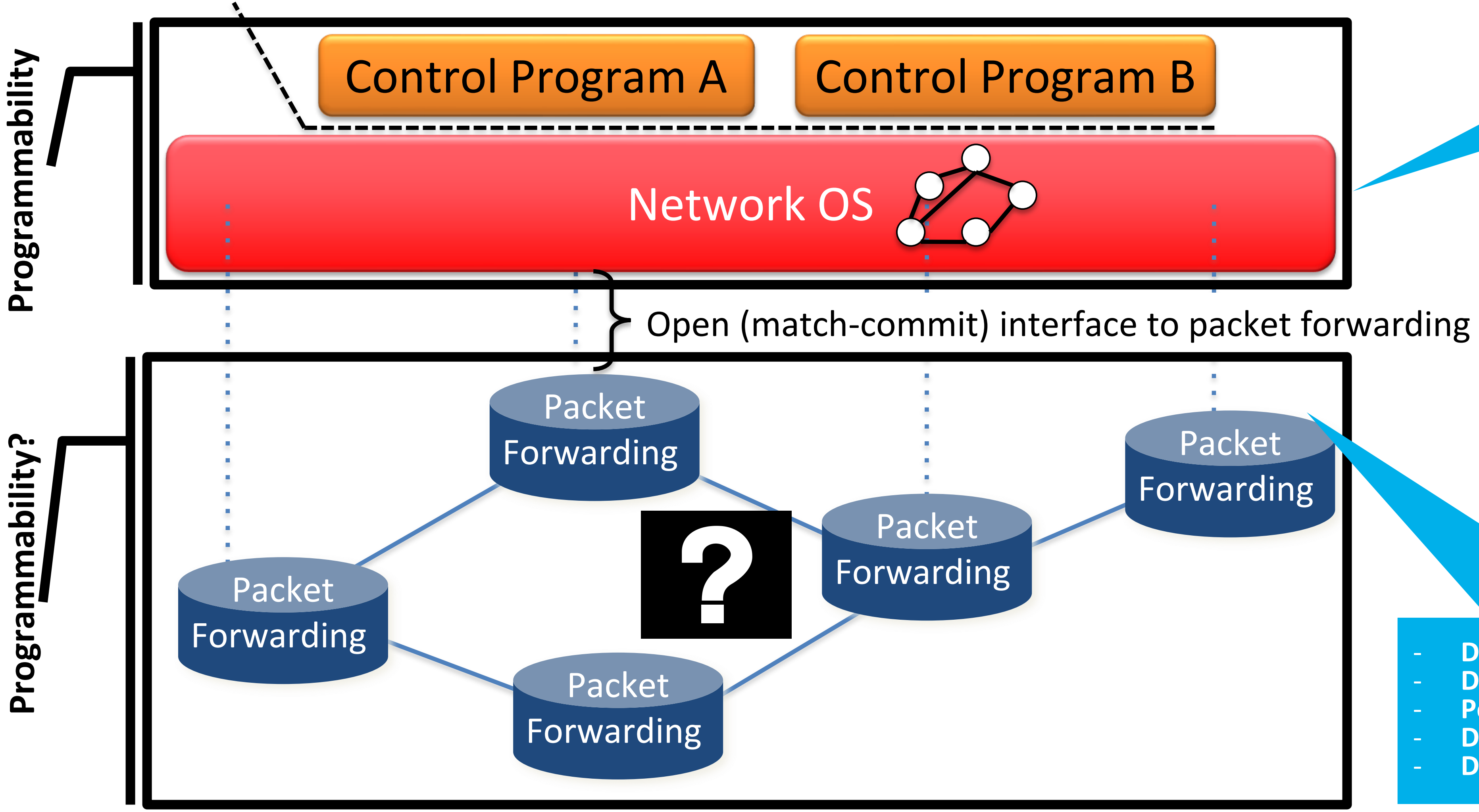
[P]: Network is Changing: Software-Defined Networking (SDN)



© Nick McKeown – Stanford University

[P]: SDN: Centralized Network-wide Decision-making

Goal: consistent, up-to-date **Global** network view



- **Dijkstra (OSPF)**
- **Spanning Tree**
- **ACL**
- **(Simple) Accounting**

- **Distributed Monitoring (telemetry)?**
- **Distributed Control?**
- **Performance-based provisioning?**
- **Distributed Anomaly Detection?**
- **Dynamic Firewalls?**

© Nick McKeown – Stanford University

[P]: Centralise Knowledge & Distribute Intelligence

Goal: Programmable Data Plane by distributing in-network processing tasks, thus, achieving programmable functionality **at line rates**

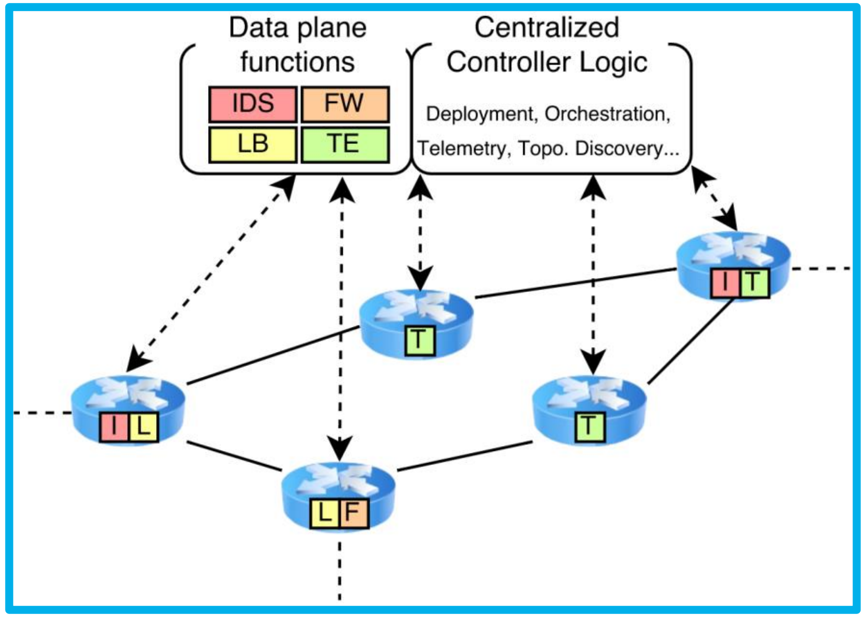
BPFabric: Central Control Logic installs Data Plane functions to the networked devices, i.e., defines their switch forwarding **behaviour** (like…AI-agents)

Per-switch behaviour: packet parsing, matching, forwarding, analytics, triggering, warning:

- Protocol-independent
- Platform-independent
- Language-independent
- Stateful – data storage, analytics, and matching

Benefit: Rapid introduction of new data plane in-network processing functions

- Routing and Forwarding; middlebox-like functions currently not possible in OpenFlow, e.g., load-balancing, telemetry, debugging, security, QoS


Data plane functions
IDS
FW
LB
TE
Centralized Controller Logic
Deployment, Orchestration, Telemetry, Topo. Discovery...
I
T
T
I
L
T
L
F

BPFabric

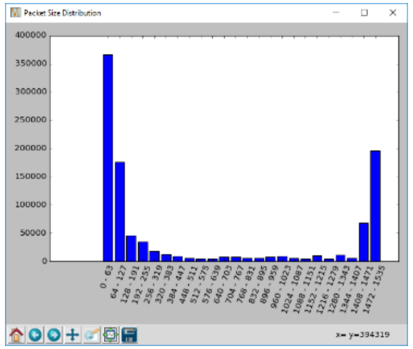
[*] BPF: Berkley Packet Filtering

[P]: BPFabric in ‘In-network Data Science’

Network Telemetry

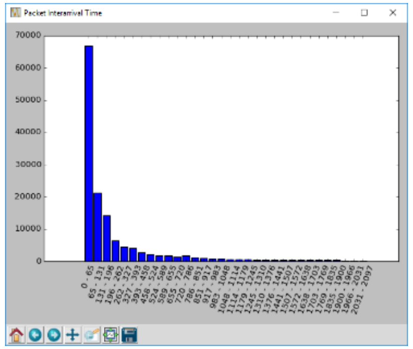
In-network Per-switch Packet-size Distribution Learning

- Controller queries the <u>current state </u>of the histogram and decides on:
  - Normal Network behaviour, i.e., classification task;
  - Traffic Trend for, e.g., latency-aware routing, latency-based optimal vNFs placement.



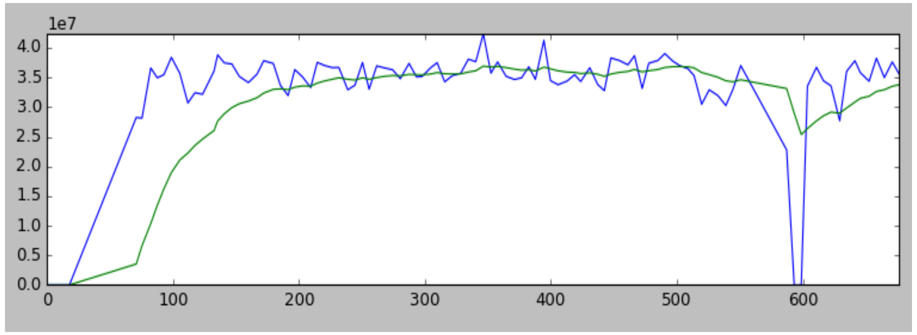
In-network Per-switch Packet inter-arrival Time Distribution Learning

- Local Learning of inter-arrival time histogram and time-of-last packet;
- Histogram pushed <u>periodically</u> to the Controller; **When?** Concept Drift SL Models…



In-network Per-switch Lightweight Anomaly Detection

- Time-series Forecast Models (e.g., ARIMA) on Incoming Traffic Volume for every port
- **Significant Deviation Rule**: Actual and Forecast Traffic <u>signals</u> the Controller; **How?** to adjust the deviation tolerance to **min.** false alarms/outliers…



NETLAB

GNF NVF Platform

NETLAB

BPFabric Data Plane

Adaptive, Resilient & Intelligent Networked Systems

means...

Time-optimized Sequential Decision Making

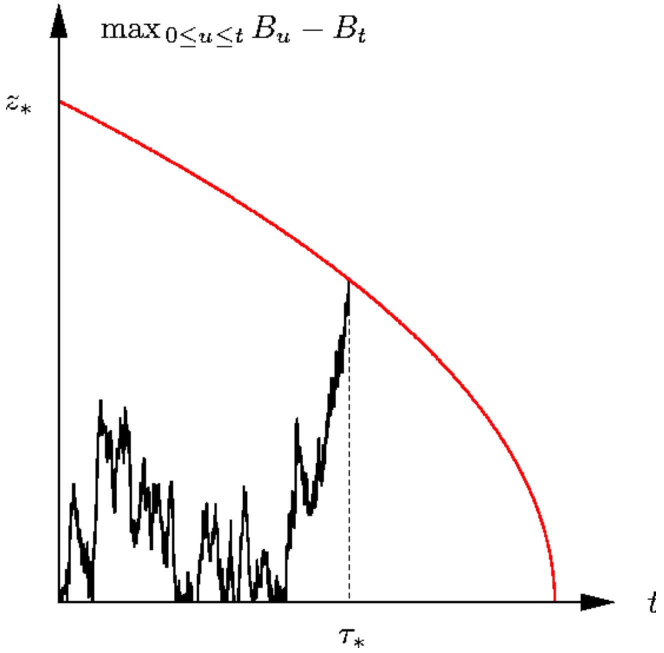
[A]: In-network Sequential Decision Making

Principles: **Optimal Stopping Theory (OST)**

- …coming from Finance; Optimal Price Selling (option trading), Secretary & Parking Problem, House Selling Problem, Casino BlackJack Problems; martingales…

Rationale: Locally Observe a (non-necessarily stationary) time-series e.g., latency values, traffic volume, contextual data streams; on e.g., Network Node, Edge Device, Sensing & Computing device, Switch, …

Task: Optimally Decide **<u>when</u>** to Process or Trigger an Action (reward/penalty/cost)



Fundamental Decisions:

- **D1:** Either **Stop and Act Now!**
- **D2:** or, …**Continue**

Objective: Find Optimal Stopping Time (stochastic optimal rule) to take an action, e.g., concept drift; anomaly detection; migrate a vNF; data transmission, **s.t.** energy/communication constraints:

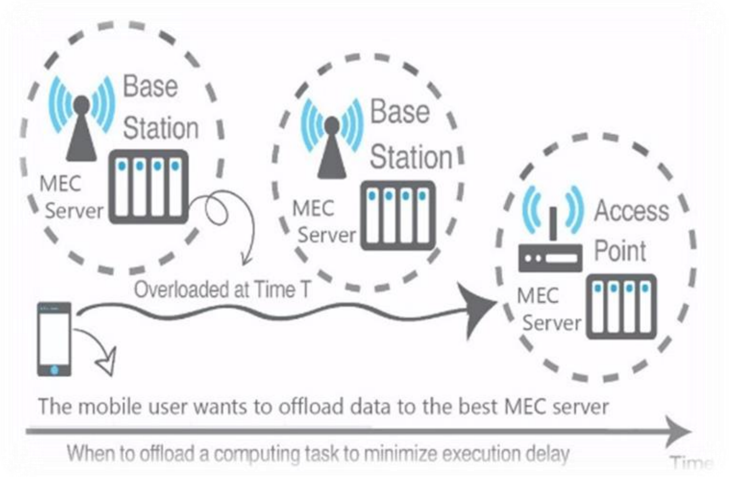
- **Maximizing Expected Reward** (e.g., best Edge Server for Task Offload)
- **Minimizing Expected Cost** (e.g., expected overall latency; outliers' false alarm; vNF migration cost due to re-placement)

[A]: Mobile Network-centric Decision Making

Goal: Time-Optimized Offloading Decision Making in Mobile Edge Computing

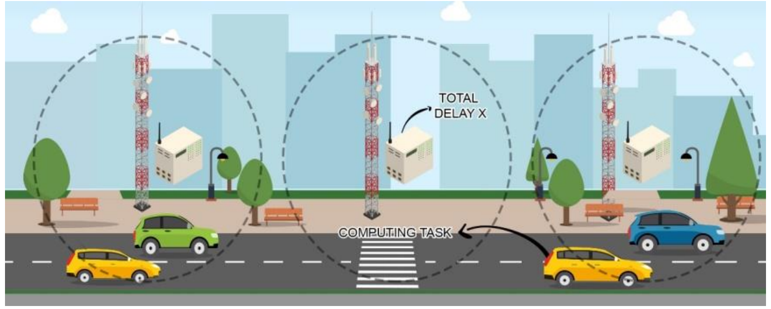
Context: Smartphone (CPU-intensive applications like 3D gaming) has data/tasks to be processed/executed and decides to offload, e.g., not available resources, limited computational capabilities.

- **Challenge 1:** When to offload tasks/data to Edge Server(s) to **min.** the expected latency/execution delay/cost [House-Selling Problem];
- **Challenge 2:** Which Edge Server(s) to offload Tasks/Data for fast processing, i.e., **max.** the probability of offloading to the 'best' server(s); [Secretary-Problem]


Base
Station
MEC
Server
Base
Station
MEC
Server
Access
Point
MEC
Server
Overloaded at Time T
The mobile user wants to offload data to the best MEC server
When to offload a computing task to minimize execution delay
Time

Constraints:

- Edge Servers' current load; User/Node mobility;
- Data Timeliness (avoid processing 'obsolete' data)
- Delay Tolerance Threshold (app specific; quality of analytics) deal with Challenges 1&2 without exceeding this (being as close as possible; [Blackjack Problem]).



[A]: Dynamic, Latency-Optimal vNF Placement at the Edge

Goal: Latency-optimal vNF placement **as close to users as** possible

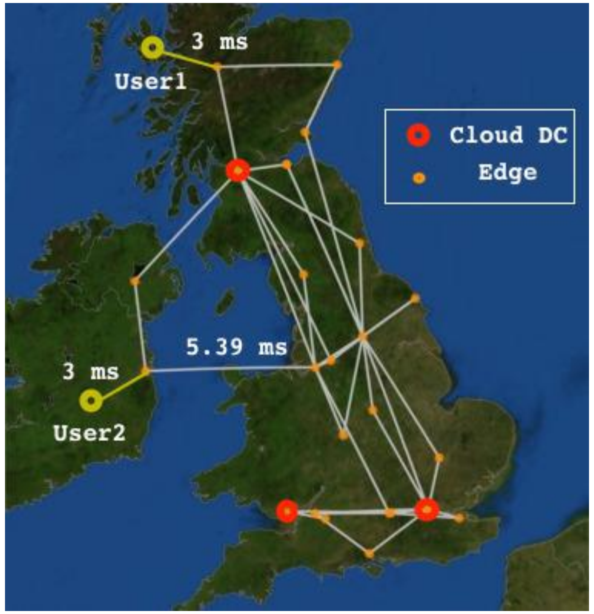
Why? Close proximity to the user implies: low latency connectivity & services at the Edge save utilization for the core network.

- **Challenge 1:** Place vNFs to **latency-optimal edge locations, i.e.,** for each vNF, find a hosting device where the 'user-to-vNF latency' is minimized.
- **Constraints**: Hardware limitations (Edge Servers); maximum tolerant latency per link; bandwidth constraints, …

Conventional Approach: e.g., ILP allocates currently vNFs to latency-optimal location.

However:

- Users move between edge devices & latencies change on links frequently!
- Users' applications impact traffic & congestion on the paths
- All impact the once original optimal allocation!



[A]: Dynamic, Latency-Optimal vNF Placement at the Edge

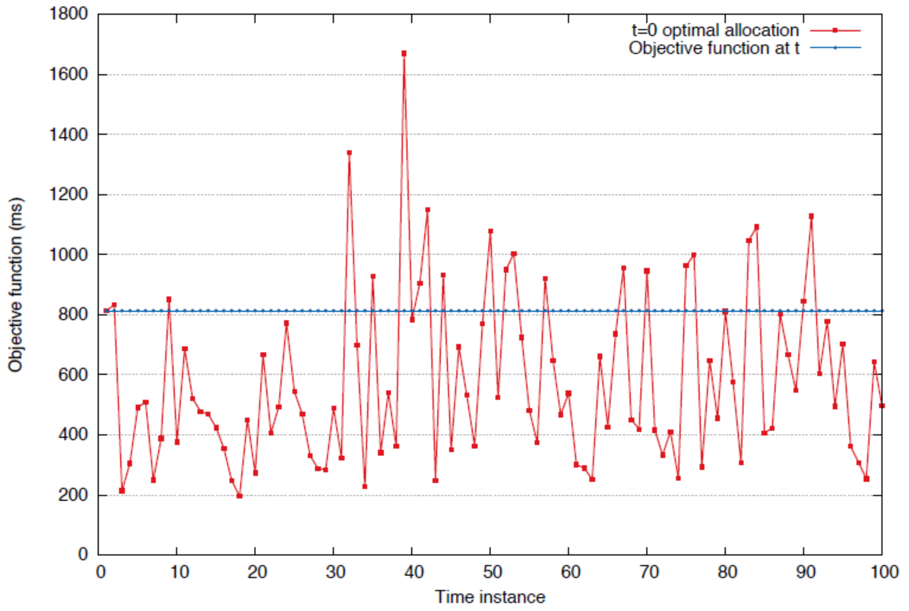
Idea: Dynamic re-allocate vNFs to keep allocation latency-optimal, i.e., re-optimize the placement with a new optimal vNFs placement …

But:

- That costs vNFs migrations & placement calculation/realization
- When is the best time for this re-optimization?

Challenge 2: Decide when to optimally re-allocate vNFs while keeping the expected number of vNFs migrations low.

- Every Time (we can): easy, always latency optimal allocation, but way too many migrations & calculations!
- Periodically: easy, require non-trivial migrations prediction, results in too many latency violations, if the period is too long…
- Cast as: Optimal Stopping Time Optimization Problem {monitoring latency and their associated migration costs [Parking-Problem]}
  - **Fact:** Low number of latency violations and low number of migrations

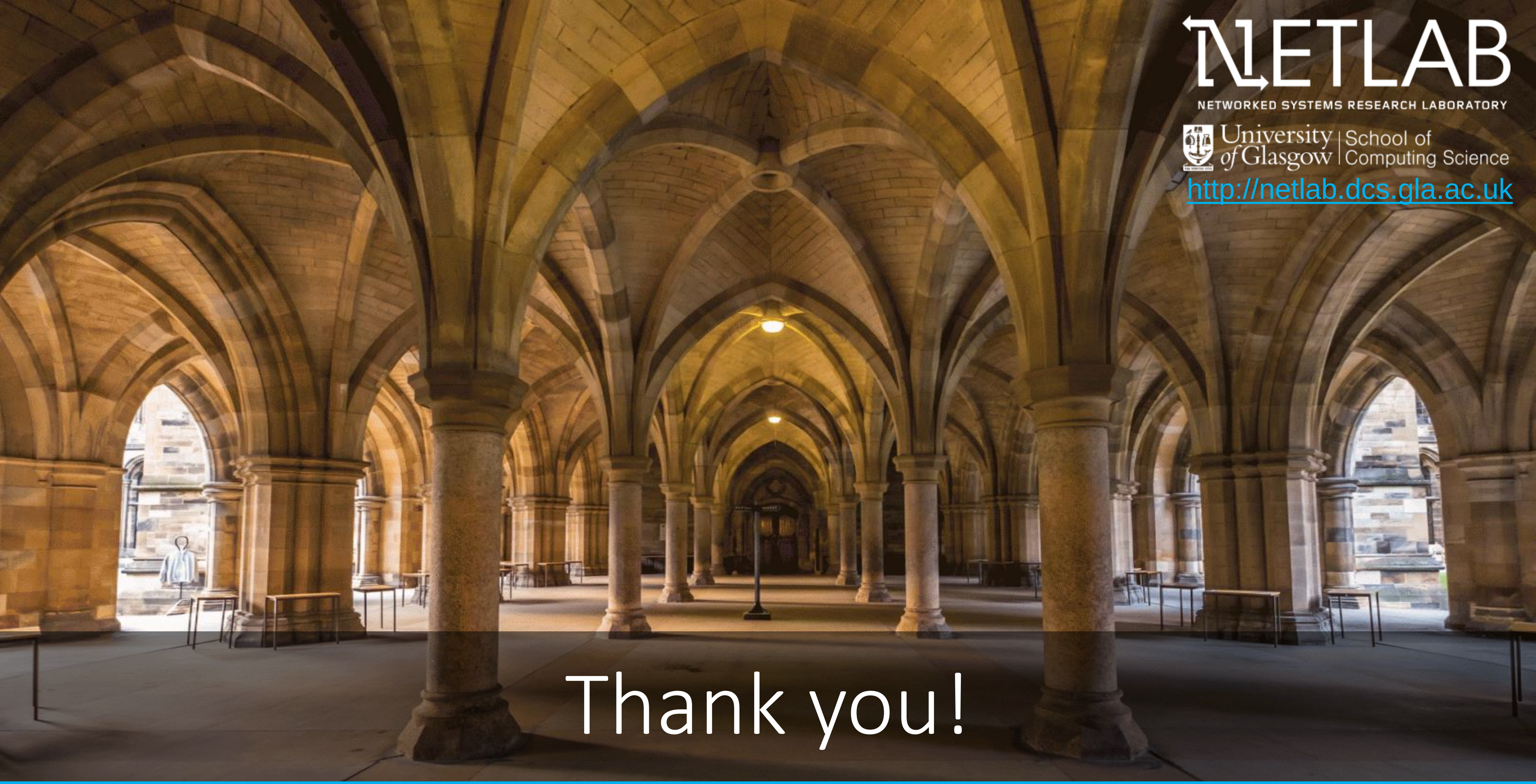


Summary

Five Dimensions in Future Networked Systems:

- Pillars: Virtualisation & Programmability via **GNF** and **BPFabric**
- **Adaptive**, **Intelligent** & **Resilient** service provisioning.
- Support **Self-*** properties based on in-network processing & data plane programmability

- Exploit infrastructure support & trends in **Optimal Stopping Theory**, **SL/ML Algorithms**, and **Bio-inspired Computing** for building resilient systems
- Advanced (value-add) services can be built on top to unleash future data communications market potential…



NETLAB
NETWORKED SYSTEMS RESEARCH LABORATORY

University of Glasgow | School of Computing Science

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

Thank you!