Network Function (NF) Parallelisation for NF-aware Traffic Distribution

Wajdi Hajji Posco Tso Dimitrios Pezaros

Coseners 2017

June 22, 2017

Loughborough University

Network Function Chaining (NFC)

A middle-box or network function is any intermediary device that transforms, inspects, filters, or otherwise manipulates traffic for purposes other than packet forwarding.

Network Function Chaining (NFC)

A middle-box or network function is any intermediary device that transforms, inspects, filters, or otherwise manipulates traffic for purposes other than packet forwarding.

Network Function Chaining (NFC) provides the ability to define an ordered list of network services. These services are then "stitched" together in the network to create a service chain (SC).

Network Function Chaining (NFC)

A middle-box or network function is any intermediary device that transforms, inspects, filters, or otherwise manipulates traffic for purposes other than packet forwarding.

Network Function Chaining (NFC) provides the ability to define an ordered list of network services. These services are then "stitched" together in the network to create a service chain (SC).

Examples of service chains in data centres

- North-South: WOC[a] : EdgeFW (e.g. VPN, NAT) : MON : ADC[b] : AppFW
- East-West: SegFW (e.g. for VLAN) : ADC : MON : AppFW

[a] Web Optimisation Control
[b] Application Delivery Controller

How has NFC performance been improved?

Network-aware orchestration layer for middle-boxes

Elastic scaling based on the bandwidth availability of network links.

How has NFC performance been improved?

Network-aware orchestration layer for middle-boxes

Elastic scaling based on the bandwidth availability of network links.

Placement optimisation

Depending on the desired goal, e.g. decrease number of forwarding rules on SDN switches, optimise the bandwidth utilisation (for instance, by locating middle-boxes on the same rack server), or reduce the middle-boxes migration overhead.

How has NFC performance been improved?

Network-aware orchestration layer for middle-boxes

Elastic scaling based on the bandwidth availability of network links.

Placement optimisation

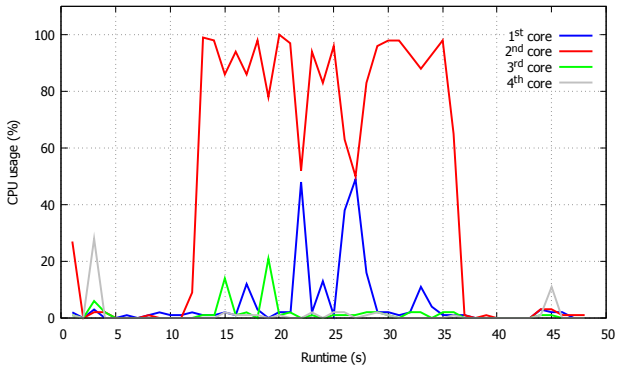
Depending on the desired goal, e.g. decrease number of forwarding rules on SDN switches, optimise the bandwidth utilisation (for instance, by locating middle-boxes on the same rack server), or reduce the middle-boxes migration overhead.

New ideas

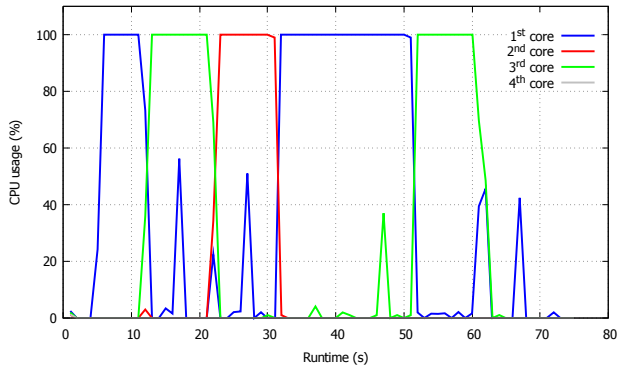
Simultaneous packet processing at parallelisable NF, where possible (Parabox).

Inside virtualised network function – Software implementation constraints?

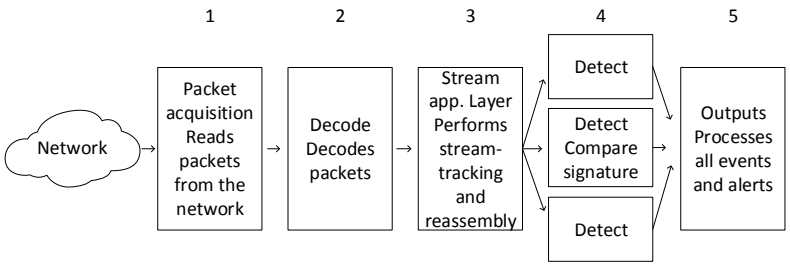
Scaling-up is not an effective way to improve the performance of single-threaded NF



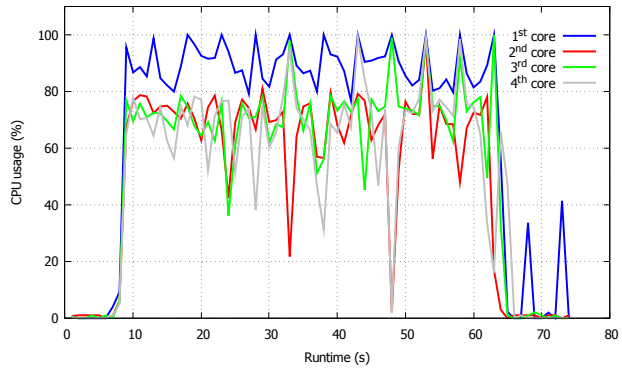
pfSense NAT, single-threading behaviour



Snort, single-threading in each phase



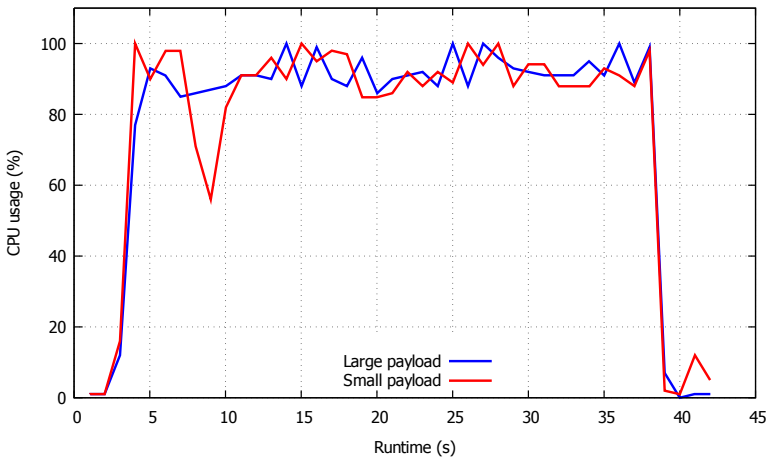
Parallel "Detect" in Suricata



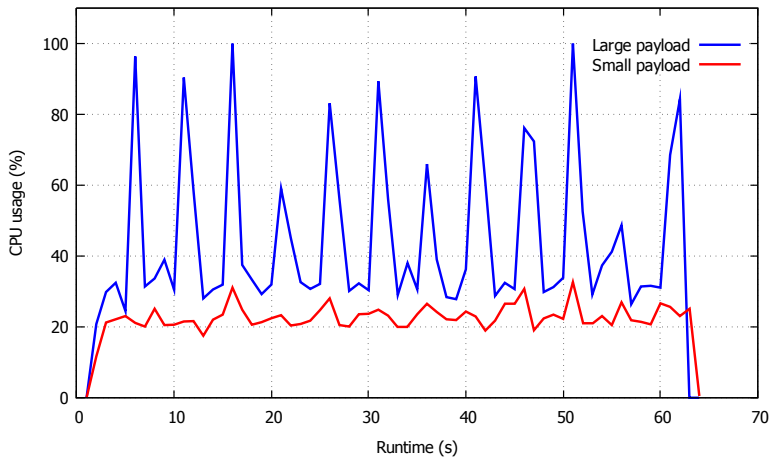
Suricata, multi-threaded NF

NF main characteristics – I/O Bound vs. Compute Bound

- R/W packet header NF performance is sensitive to the packet rate, and in particular vulnerable at a high rate
- Performance of NF dealing with packet payload is sensitive to the throughput (packet rate and payload)



Payload size effect on pfSense NAT



Payload size effect on Snort (IDS)

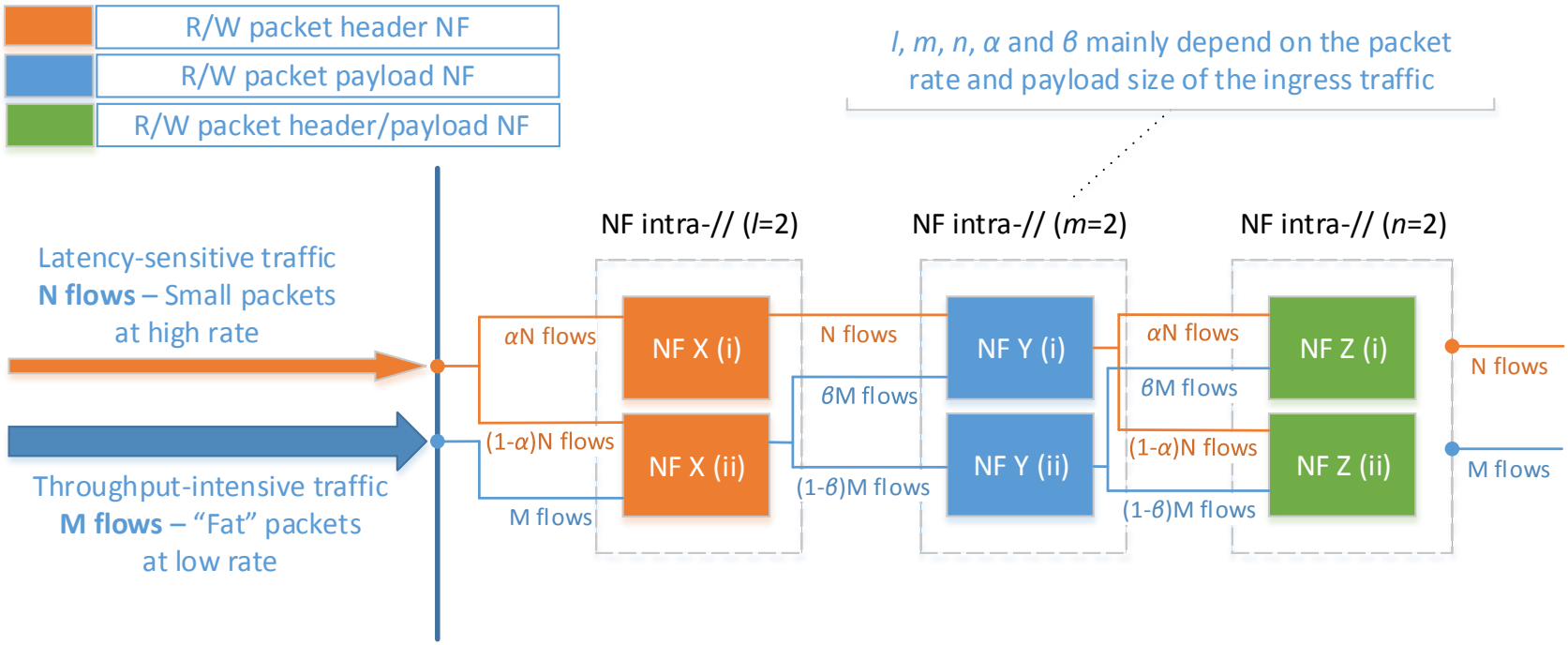
NAT and IDS show different responses towards packet payload size

Revise existing NF instantiation module

Cyclehoop – cycle parking service

Fine-tuning the traffic management scheme within the SC

Interference and concurrence between latency-sensitive and data-intensive application traffics, the latter causes congestion to the former.
Hadoop, search engines, trading platforms as examples.

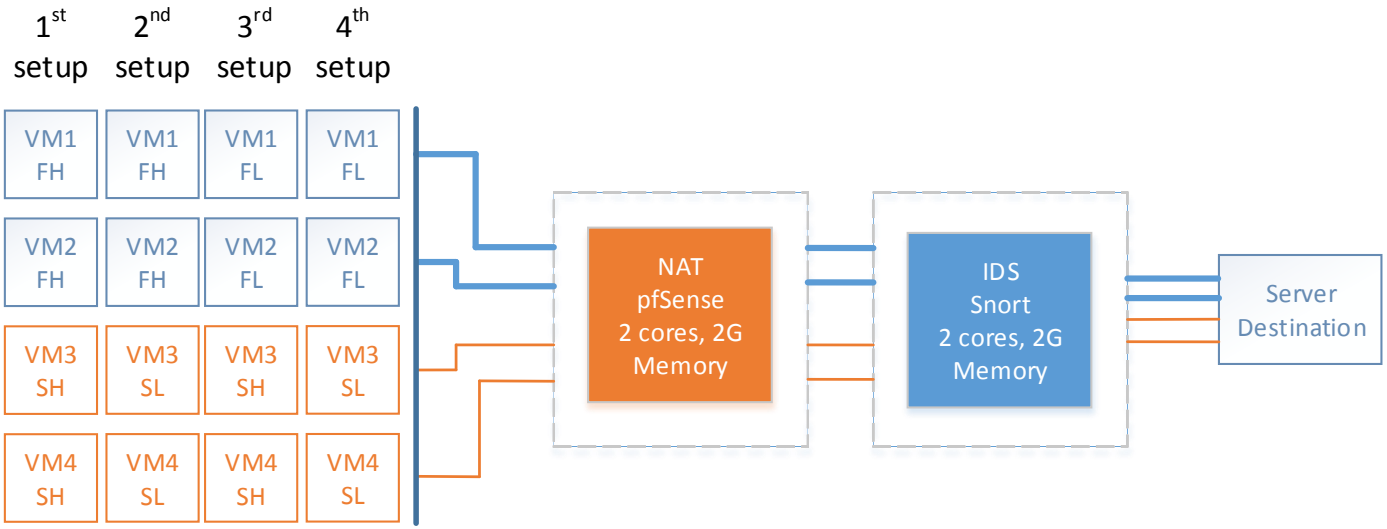


A particular case of flow-based traffic forwarding plan within a SC of three NF

Baseline experiment – SC traditional set-up

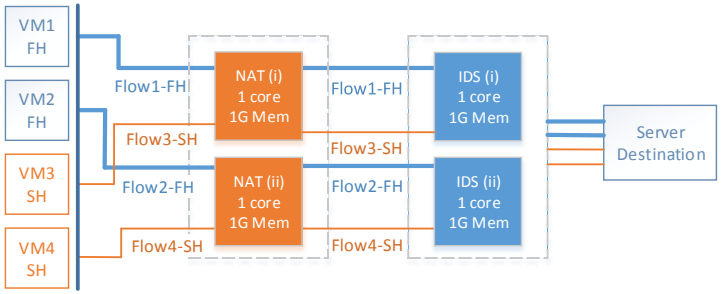
In each run, we apply a set of flows (traffic set-up) through the SC

- Large (**F**) pkts at **H**igh rate & **S**mall pkts at **H**igh rate: FHSH
- Large (**F**) pkts at **H**igh rate & **S**mall pkts at **L**ow rate: FHSL
- Large (**F**) pkts at **L**ow rate & **S**mall pkts at **H**igh rate: FLSH
- Large (**F**) pkts at **L**ow rate & **S**mall pkts at **L**ow rate: FLSL

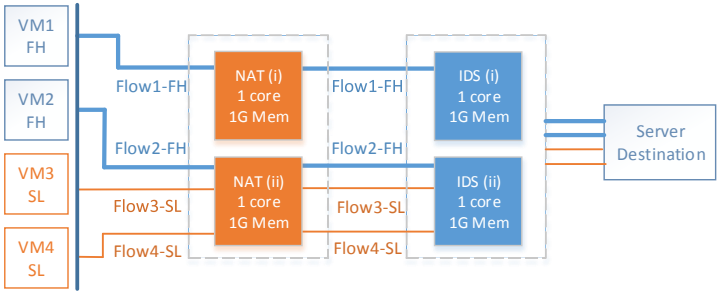


A service chain of two virtualised network functions; pfSense NAT and Snort IDS – both at 2 cores and 2G memory

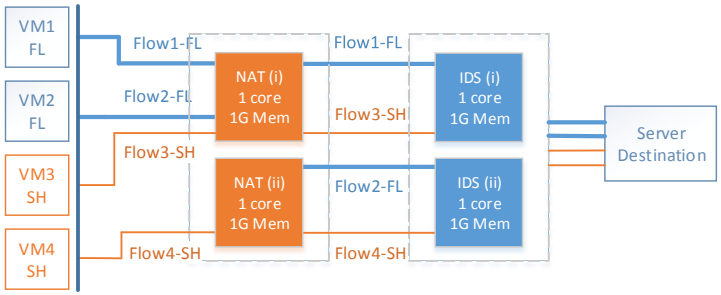
Proof-of-concept experiment



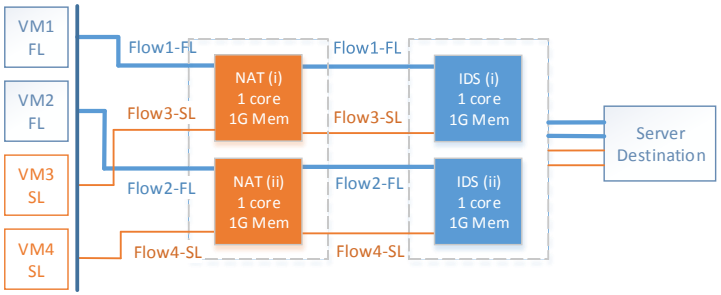
1st set-up – FHSH



2nd set-up – FHSL



3rd set-up – FLSH

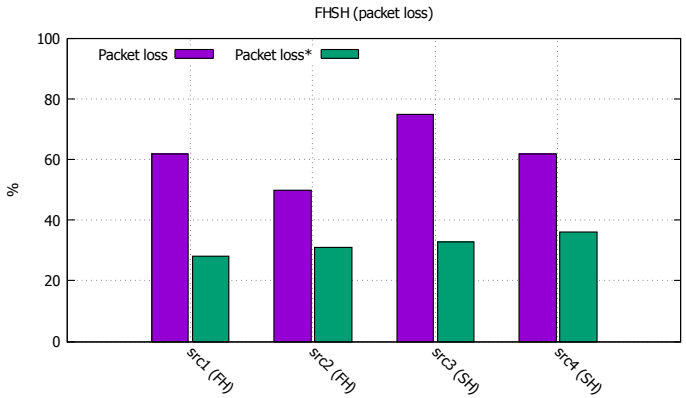


4th set-up – FLSL

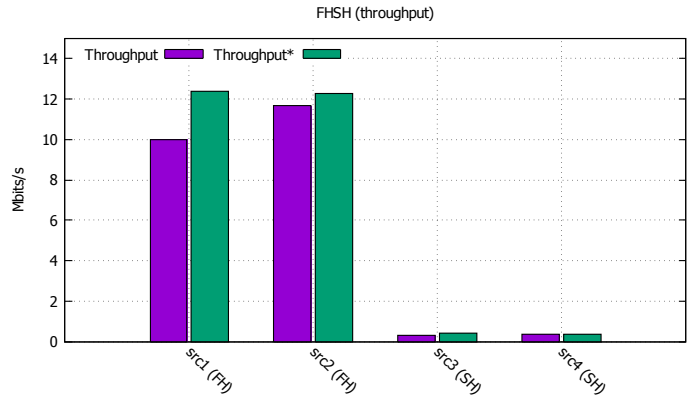
Adapting traffic distribution to the SC ingress traffic nature, and resources reduced by half for each NF instance

Preliminary results

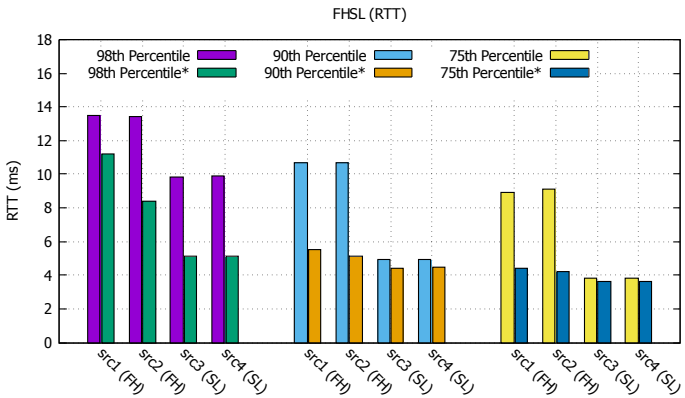
Packet loss mitigated by 75.86%, latency by 21.74%, and throughput grew by 8.28%



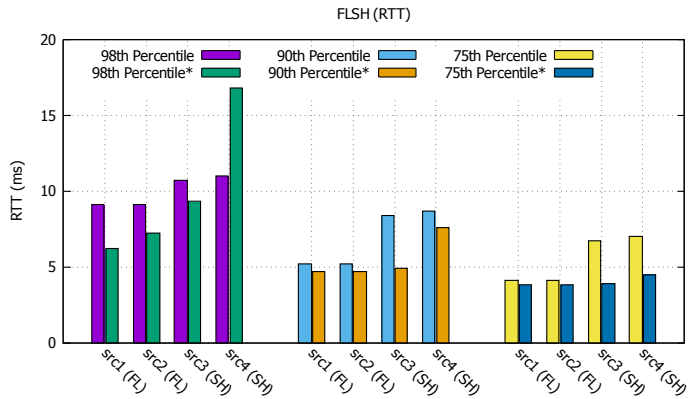
Packet loss dropped



Throughput increased



Latency reduced for FH



..and also for SH traffic

Conclusion and next step

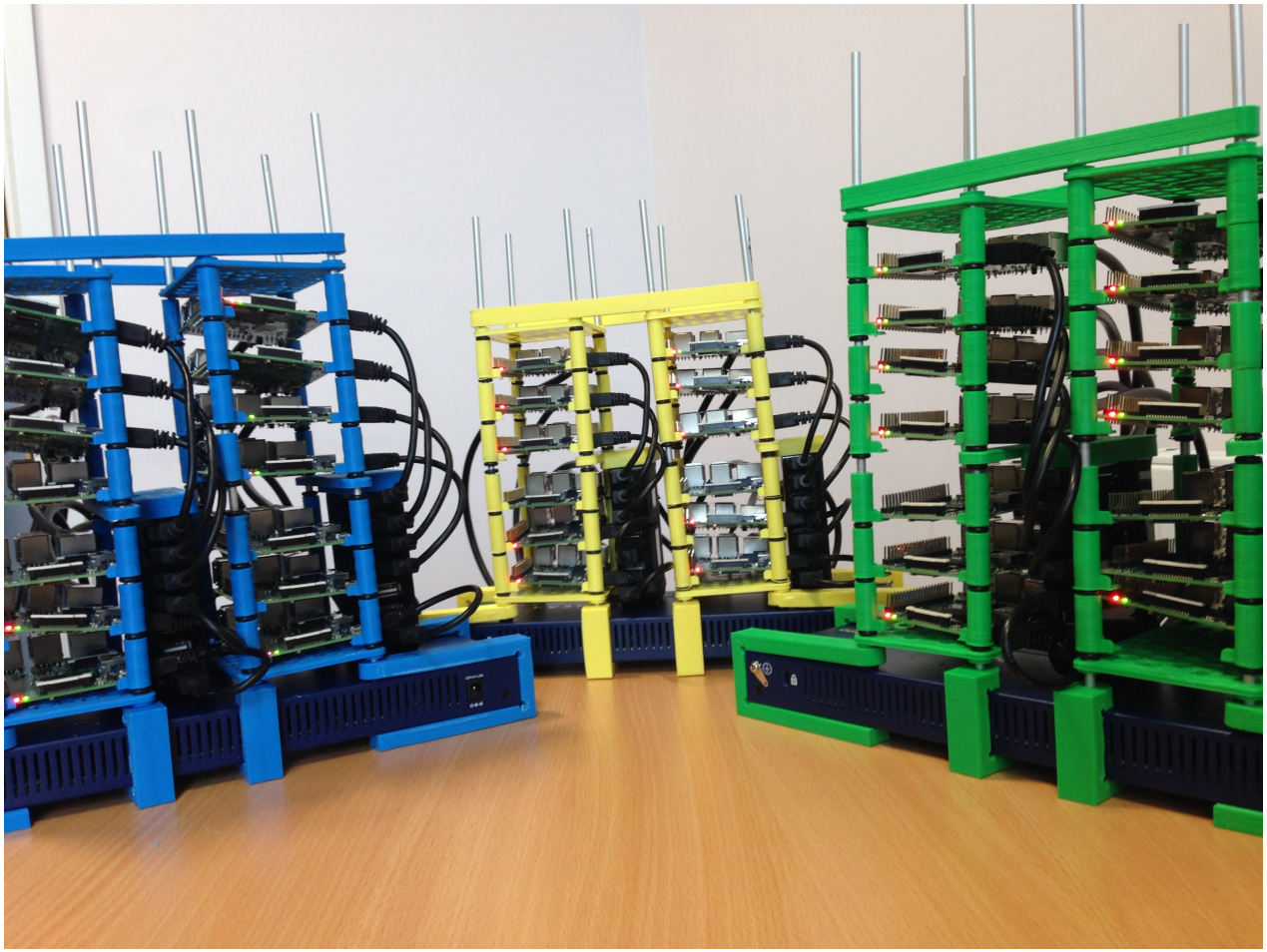
- Considering the implementation characteristics of the NF can help to optimise the resources allotment (single-threading vs. multi-threading).
- NF differently deals with packets (I/O Bound vs. Compute Bound).
- Flow-based traffic distribution is essential for stateful network functions (to ensure its proper functioning).

Conclusion and next step

- Considering the implementation characteristics of the NF can help to optimise the resources allotment (single-threading vs. multi-threading).
- NF differently deals with packets (I/O Bound vs. Compute Bound).
- Flow-based traffic distribution is essential for stateful network functions (to ensure its proper functioning).

- Mathematically model the problem to find out the optimum values of l, m, n, α, and β (remember? Slide No 8).
- Testbed evaluation of the eventual algorithms' efficiency in terms of resources consumption and execution time.

What else? Deployment on Raspberry Pi devices – NF in IoT?



Miniaturised data centre made of Raspberry Pi

Thank you! Questions?