

Disaggregated Data Centre Architecture: Switching, Networking & Re-programmability

Vaibhava Mishra, Georgios Zervas

Optical Network Group

Department of EEE

University College London



Outline

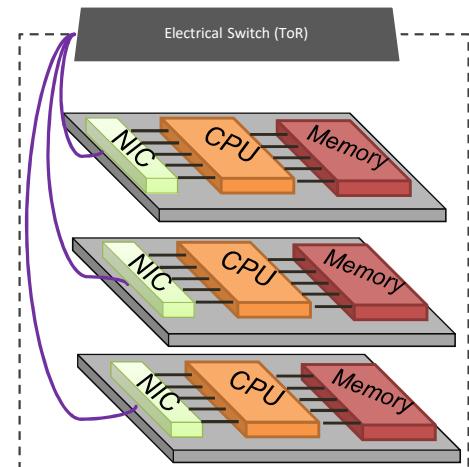
- Disaggregated Data Centre architectures
- **dReDBox EU H2020 project**
- dRedBox architecture and technologies
- Virtual Function Service Chain: Protocol Independent Switch
- Reconfiguration in disaggregated architecture



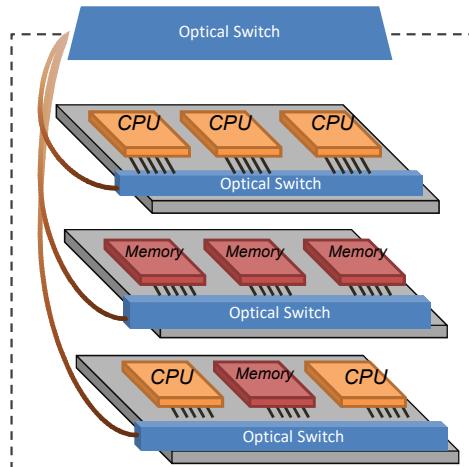
2

Ultimate Disaggregation in Data Centres

Server-Centric Model
Traditional Rack with direct-attached memory
Electrical interconnect



Function-Centric Disaggregated Model
Rack with network-attached memory
Optical Interconnect



dReD
Box

3

Why disaggregated Data Centres?

"Data Centres often use less than **40%** of their available computing resources"

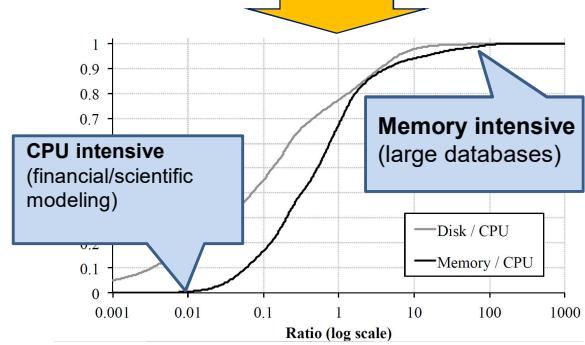
Disaggregated resources
(processing, DRAM, storage)



- **modularity**
- **Cost and power**
- **Upgradeability**
- **Utilization** of resources

Distribution of relative storage/memory capacity demand to CPU usage for tasks in Google's DC

The resource requirements of tasks are spread over more than three orders of magnitude!



4



dReDBox consortium

- Project details
 - **Contract number:** INFSO-ICT-687632
 - **Coordinator:** IBM Research - Ireland, katrinisk@ie.ibm.com
 - **Community contribution:** 6.5M€
 - **Start date:** January 1st, 2016
 - **Duration:** 36 Months



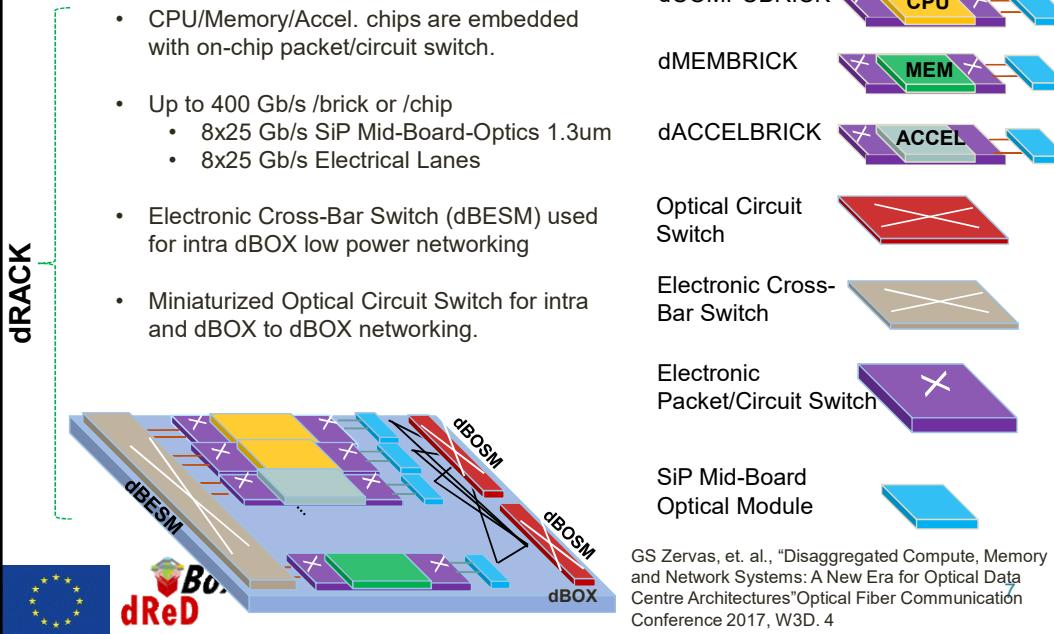
5

dReDBox consortium

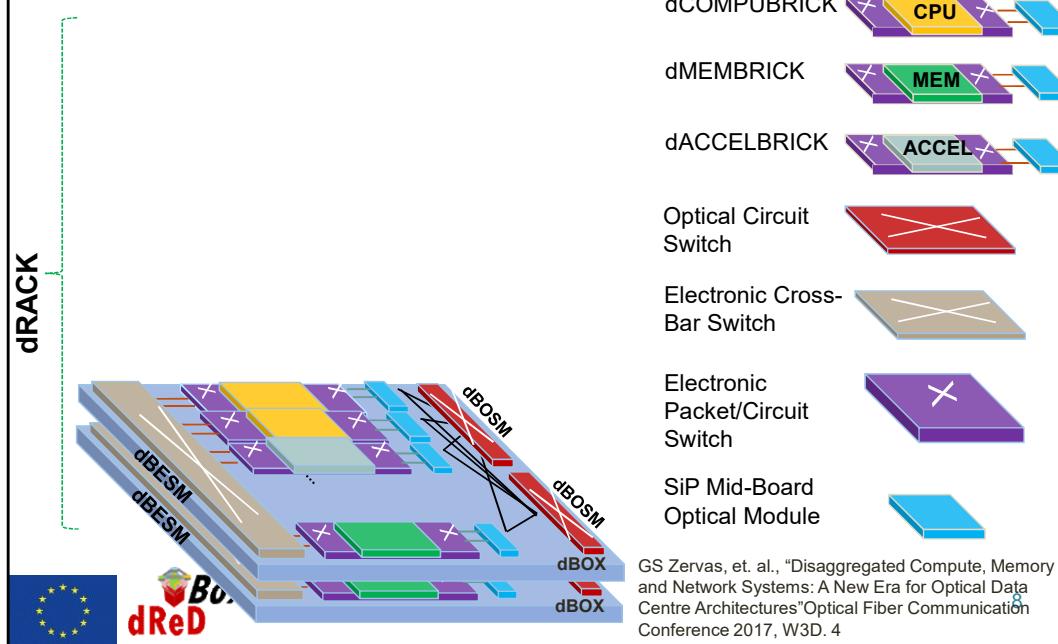
- Project consortium of 11 partners



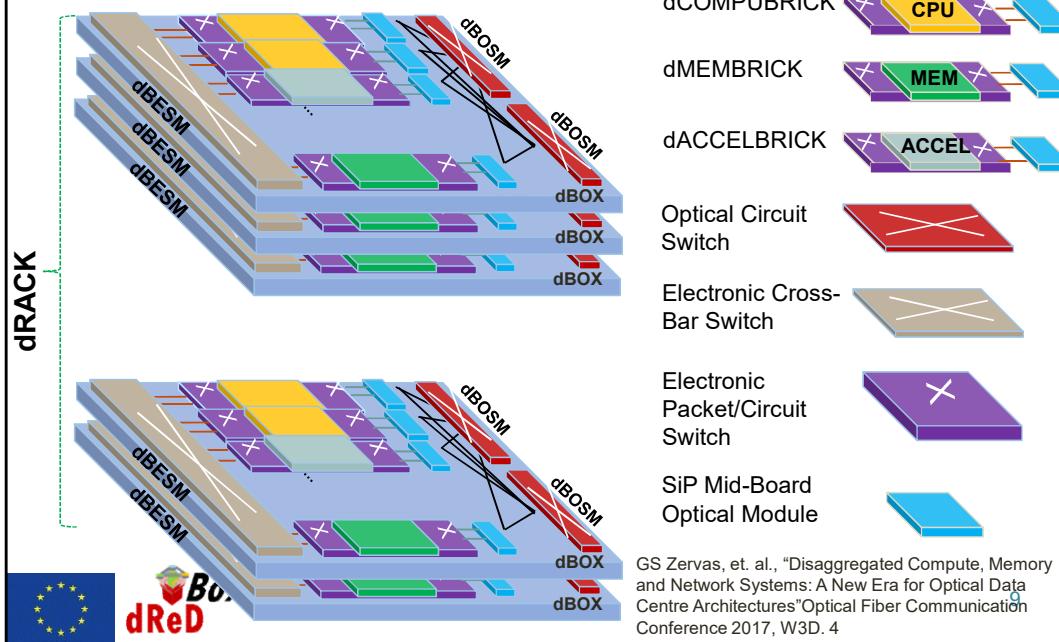
dReDBox dRACK Architecture



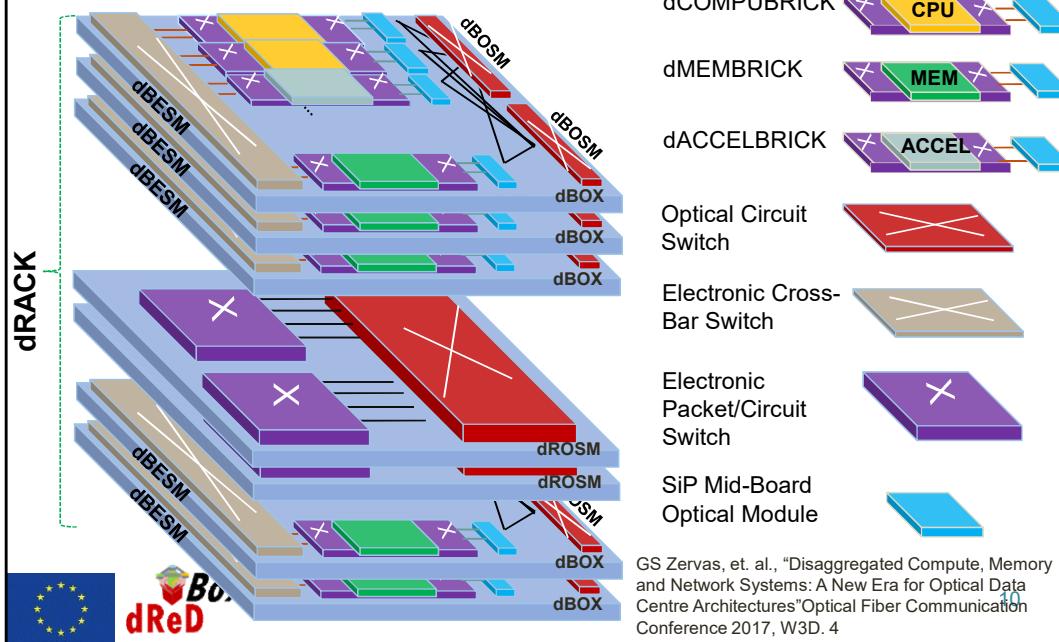
dReDBox dRack Architecture



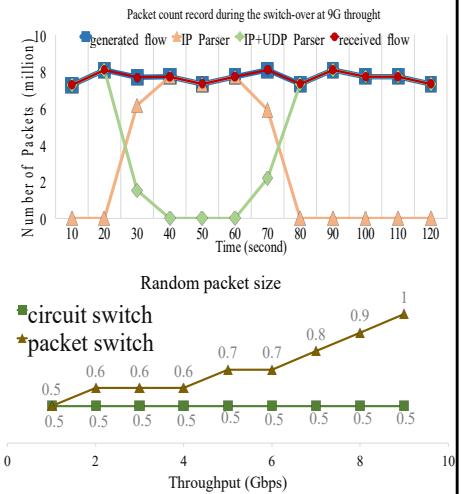
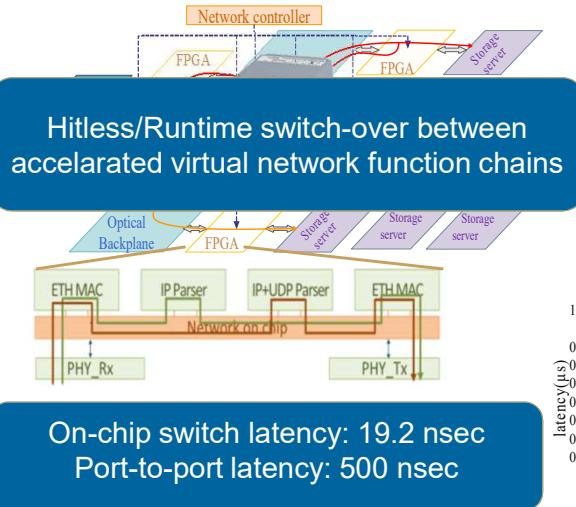
dReDBox dRack Architecture



dReDBox dRack Architecture



Virtual Function Service Chain : Protocol Independent Switch



Chen, Q., Mishra, V., Zervas, G., "Reconfigurable Computing for Network Function Virtualization: A Protocol Independent Switch", IEEE Reconfig 2016, , Nov-Dec 2016

11

Optical Networks

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates the internal structure of an FPGA for partial reconfiguration. It is organized into several horizontal layers:

- FPGA**: The outermost layer.
- Reconfigurable Region**: A yellow layer containing a **Network Function**.
- Network on Chip**: An orange layer positioned below the Network Function.
- Blank Reconfigurable Region**: A yellow layer at the bottom.

A teal arrow points from the left towards the Network Function, indicating the path of a signal or configuration update. Another teal arrow points from the Network Function towards the right, indicating the output or further propagation of the signal.

Optical Networks

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates a Field-Programmable Gate Array (FPGA) structure. At the top, a light blue bar contains the text "Optical Networks" on the left and the "UCL" logo on the right. Below this is a white header bar with the title "Reconfigurable Network: Partial Reconfiguration". The main content area shows an "FPGA" block. Inside the FPGA, there is a "Reconfigurable Region" highlighted in yellow. This region contains a "Network Function" block. Below the reconfigurable region is a "Network on Chip" block, which is orange. At the bottom of the FPGA is a "New Network Function" block, which has a dotted pattern. A green arrow points from the "Run time Partial re-configuration" text in a speech bubble to the boundary between the reconfigurable region and the network on chip.

FPGA

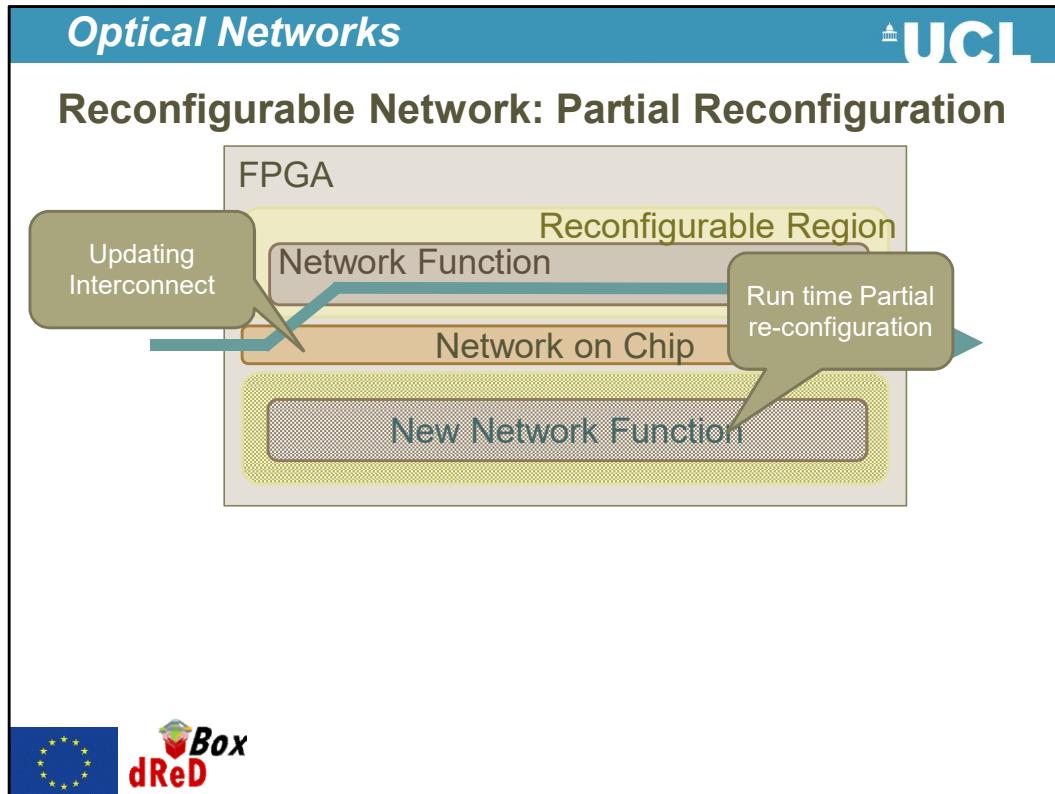
Reconfigurable Region

Network Function

Network on Chip

New Network Function

Run time Partial re-configuration



Optical Networks

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates a cross-section of an FPGA (Field-Programmable Gate Array). It shows three distinct horizontal layers. The top layer is labeled "Inactive" and has a yellow diamond pattern. The middle layer is labeled "Network on Chip" and is orange. The bottom layer is labeled "New Network Function" and is yellow. A teal arrow points from the left towards the "Network on Chip" layer, and another teal arrow points from the "Network on Chip" layer towards the right.

Optical Networks

UCL

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates the concept of partial reconfiguration in an FPGA. It shows a large grey box labeled 'FPGA' containing three horizontal sections. The top section is grey with a diamond pattern and labeled 'Inactive'. The middle section is orange and labeled 'Network on Chip'. The bottom section is yellow and labeled 'New Network Function'. A teal arrow points from the left towards the 'Network on Chip' section, indicating the area being targeted for reconfiguration. Another teal arrow points from the right towards the 'New Network Function' section, indicating the area being added or modified.

FPGA

Inactive

Network on Chip

New Network Function

No to Static reconfiguration

Avoid on/off device

Optical Networks

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates the concept of partial reconfiguration in an FPGA. It shows a large grey box labeled 'FPGA' containing three horizontal sections. The top section is grey with a diagonal hatching pattern and is labeled 'Inactive'. The middle section is orange and labeled 'Network on Chip'. The bottom section is yellow and labeled 'New Network Function'. A teal arrow points from the text 'No to Static reconfiguration' towards the 'Inactive' section. Another teal arrow points from the text 'No to packet loss' towards the 'New Network Function' section. To the right of the 'Network on Chip' section, a teal arrow points towards the text 'Avoid on/off device'. Below the 'New Network Function' section, a teal arrow points towards the text 'Hitless switch over'.

FPGA

Inactive

Network on Chip

New Network Function

No to Static reconfiguration

Avoid on/off device

No to packet loss

Hitless switch over

Optical Networks

UCL

Reconfigurable Network: Partial Reconfiguration

The diagram illustrates the internal structure of an FPGA during partial reconfiguration. It consists of three stacked horizontal regions:

- Inactive:** The top region is grey with a diamond pattern.
- Network on Chip:** The middle region is orange.
- New Network Function:** The bottom region is yellow.

A teal arrow points from the left towards the 'Network on Chip' region, indicating the area being modified or reconfigured. Another teal arrow points from the right towards the 'New Network Function' region, indicating the new function being implemented.

No to Static reconfiguration	Avoid on/off device
No to packet loss	Hitless switch over
Partial bit files challenges : Storage, delivery	Defining re-configuration protocol

Optical Networks

Reconfiguration in Disaggregated Architecture : Challenges

Remote configuration

Reconfiguration (Remote Bricks)

Optical Networks

Reconfiguration in Disaggregated Architecture : Challenges

The diagram illustrates the components of a disaggregated architecture. At the top left is a cloud-like shape containing the text "Optimized Less Resource". At the top right is another cloud-like shape containing the text "Remote configuration". In the center is a yellow oval containing the text "Reconfiguration (Remote Bricks)". Three small brown dots connect the three main components, forming a triangle.

Optical Networks

Reconfiguration in Disaggregated Architecture : Challenges

Optimized Less Resource

Remote configuration

Reconfiguration (Remote Bricks)

Number of dBricks

Optical Networks

Reconfiguration in Disaggregated Architecture : Challenges

The diagram illustrates the challenges of reconfiguration in a disaggregated architecture. At the center is a yellow oval labeled "Reconfiguration (Remote Bricks)". Surrounding it are four cloud-shaped bubbles: "Optimized Less Resource" (top left), "Remote configuration" (top right), "Independent to application" (bottom right), and "Number of dBricks" (bottom left). Small brown dots connect the central oval to each of the four clouds.

Optimized Less Resource

Remote configuration

Reconfiguration (Remote Bricks)

Independent to application

Number of dBricks

Optical Networks

UCL

Reconfiguration in Disaggregated Architecture : Challenges

The diagram illustrates the challenges of reconfiguration in a disaggregated architecture. At the center is a yellow oval labeled "Reconfiguration (Remote Bricks)". Surrounding it are six cloud-shaped boxes, each containing a challenge: "Optimized Less Resource" (green), "Reliability" (orange), "Remote configuration" (green), "Independent to application" (green), "Number of dBricks" (orange), and "Optimized Less Resource" (green). Small brown dots connect the central oval to each of the surrounding clouds.

Optimized Less Resource

Reliability

Remote configuration

Reconfiguration (Remote Bricks)

Number of dBricks

Independent to application

Optical Networks

UCL

Reconfiguration in Disaggregated Architecture : Challenges

The diagram illustrates the challenges of reconfiguration in a disaggregated architecture. At the center is a yellow oval labeled "Reconfiguration (Remote Bricks)". Surrounding it are six cloud-shaped boxes, each containing a challenge. Small brown dots connect the central oval to each of the surrounding clouds.

- Optimized Less Resource
- Reliability
- Remote configuration
- Number of dBricks
- Manageable by Network admin
- Independent to application

Optical Networks

Reconfiguration in Disaggregated Architecture : Challenges

Resource < 2%

Reliable channel to deliver bitfile

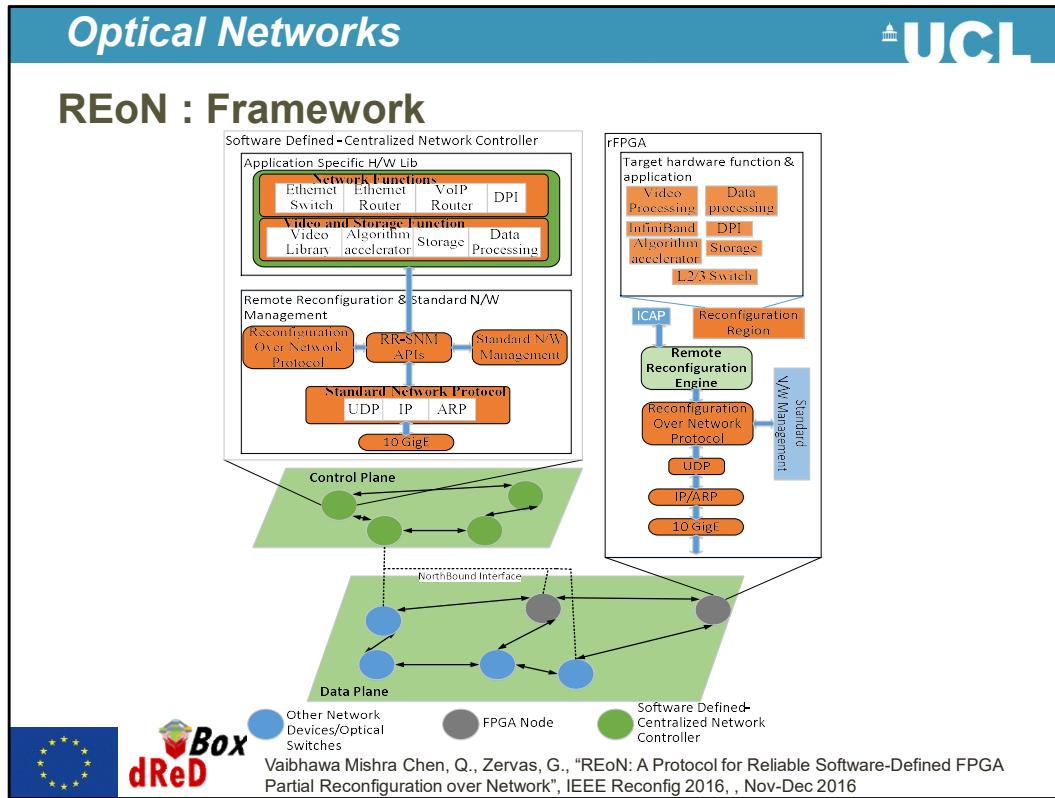
UDP/IP

Scalable

Simple API

Control management port

REoN

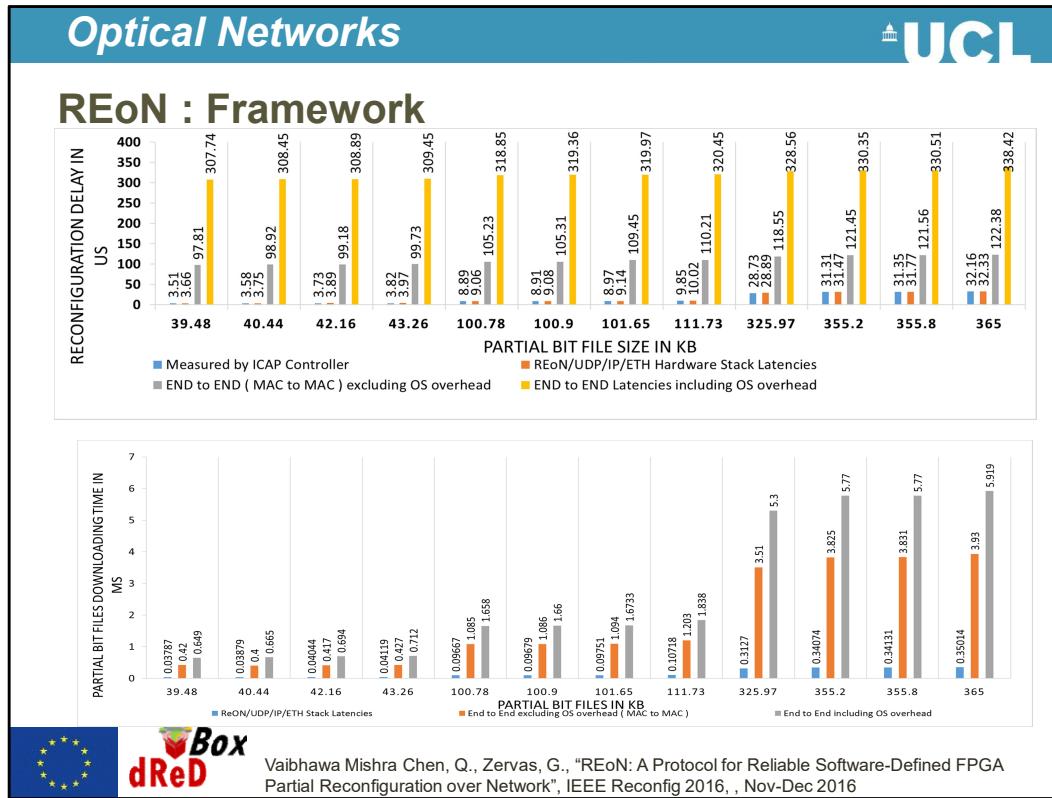


Framework allows brick as a standalone resource. It is capable to deploy new physical or virtual network functions, data and signal accelerators, on demand over the network. Our approach also advocates custom hardware network stacks, which can be very fast and resource efficient. The centralized server is used to store the partial bit files as well as reconfiguration sequence by which partial bit files can be transferred directly to network enabled FPGAs.

A high level architecture for the proposed framework is depicted. Implementation is divided in two aspects, the software part, running over the SD-CNC and hardware part, deployed on rFPGA which is nothing but the brick platforms. The SD-CNC runs the Remote Reconfiguration and Standard Network Management (RRSNM) System. The RR-SNM system is able to call an application specific function from a library through its software APIs and transfer its corresponding partial bit file to the required brick. These APIs, as the part of the RR-SNM

system have been provided to abstract low-level details of proposed protocol and provide user interaction to initiate configuration sequence. To do so, the RR-SNM APIs communicate with the Remote Reconfiguration Engine (RRE) on each brick via REoN protocol; a connection-oriented protocol over UDP which creates and reconfiguration session and guarantees order and delivery of partial bit file packets sent to the bricks from the SD-CNC.

Here the some performances measured with real time application.



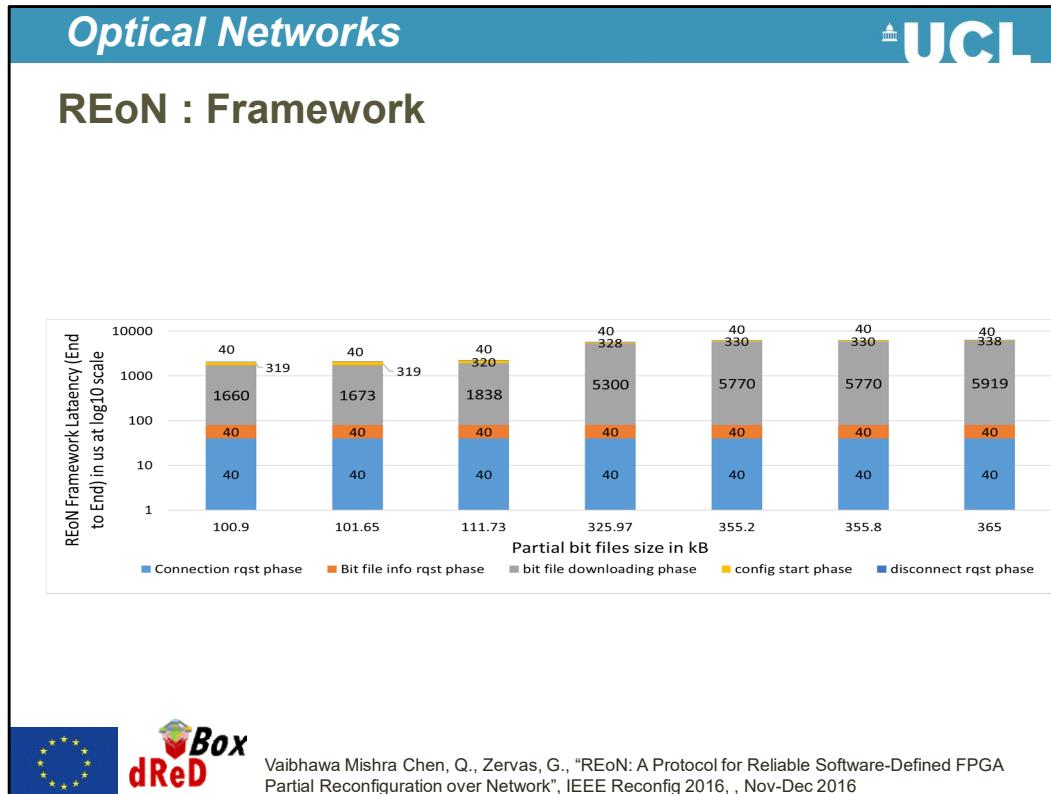
Framework allows brick as a standalone resource. It is capable to deploy new physical or virtual network functions, data and signal accelerators, on demand over the network. Our approach also advocates custom hardware network stacks, which can be very fast and resource efficient. The centralized server is used to store the partial bit files as well as reconfiguration sequence by which partial bit files can be transferred directly to network enabled FPGAs.

A high level architecture for the proposed framework is depicted. Implementation is divided in two aspects, the software part, running over the SD-CNC and hardware part, deployed on rFPGA which is nothing but the brick platforms. The SD-CNC runs the Remote Reconfiguration and Standard Network Management (RRSNM) System. The RR-SNM system is able to call an application specific function from a library through its software

APIs and transfer its corresponding partial bit file to the required brick. These APIs, as the part of the RR-SNM

system have been provided to abstract low-level details of proposed protocol and provide user interaction to initiate configuration sequence. To do so, the RR-SNM APIs communicate with the Remote Reconfiguration Engine (RRE) on each brick via REoN protocol; a connection-oriented protocol over UDP which creates and reconfiguration session and guarantees order and delivery of partial bit file packets sent to the bricks from the SD-CNC.

Here the some performances measured with real time application.



Framework allows brick as a standalone resource. It is capable to deploy new physical or virtual network functions, data and signal accelerators, on demand over the network. Our approach also advocates custom hardware network stacks, which can be very fast and resource efficient. The centralized server is used to store the partial bit files as well as reconfiguration sequence by which partial bit files can be transferred directly to network enabled FPGAs.

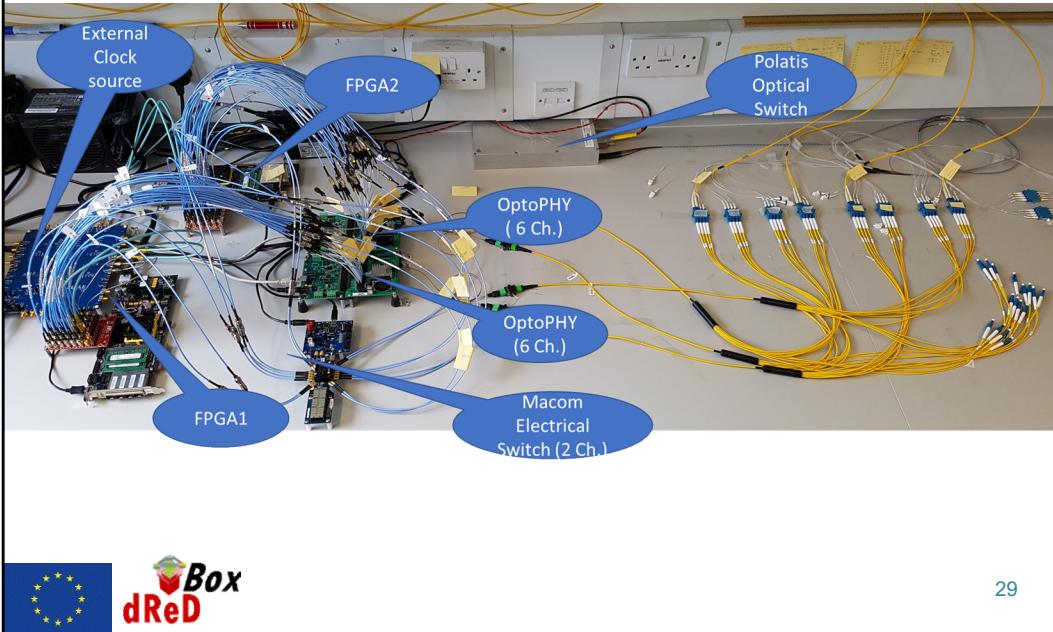
A high level architecture for the proposed framework is depicted. Implementation is divided in two aspects, the software part, running over the SD-CNC and hardware part, deployed on rFPGA which is nothing but the brick platforms. The SD-CNC runs the Remote Reconfiguration and Standard Network Management (RRSNM) System. The RR-SNM system is able to call an application specific function from a library through its software

APIs and transfer its corresponding partial bit file to the required brick. These APIs, as the part of the RR-SNM

system have been provided to abstract low-level details of proposed protocol and provide user interaction to initiate configuration sequence. To do so, the RR-SNM APIs communicate with the Remote Reconfiguration Engine (RRE) on each brick via REoN protocol; a connection-oriented protocol over UDP which creates and reconfiguration session and guarantees order and delivery of partial bit file packets sent to the bricks from the SD-CNC.

Here the some performances measured with real time application.

Testbed



 **dReD**

29

Conclusions

- dReDBox architecture can offer **optimum IT resource utilization**
- dReDBox **topology and function reconfiguration** can offer substantial benefits to classic static hybrid packet/circuit network architectures
- **Protocol independent reconfigurable computing** and network systems deliver unprecedented flexibility to support multiple service on same infrastructure.
- **REoN** can increase the network function adoptability in run time.
- Challenges:
 - Latency overheads kept to minimum
 - Cost
 - Power
 - Memory Management



30

ThanQ



31