Dr. Jin Wei-Kocsis

Assistant Professor, Electrical & Computer Engineering

The University of Akron

This research work was supported by NASA under Grant 80NSSC17K0530.

A Blockchain-Enabled Multi-Controller SDN-based Resilient Networking Management Mechanism

The University of Akron
College of Engineering

Outline

- Background & Motivations
- Our Proposed Networking Solution with Case Studies
  - Integrity Verification Mechanism
  - Autonomous Malicious-Host Detection Mechanism
  - Autonomous Bandwidth Provisioning Mechanism
- Conclusions

Background & Motivations

- Software-Defined Networking (SDN) is a promising technology

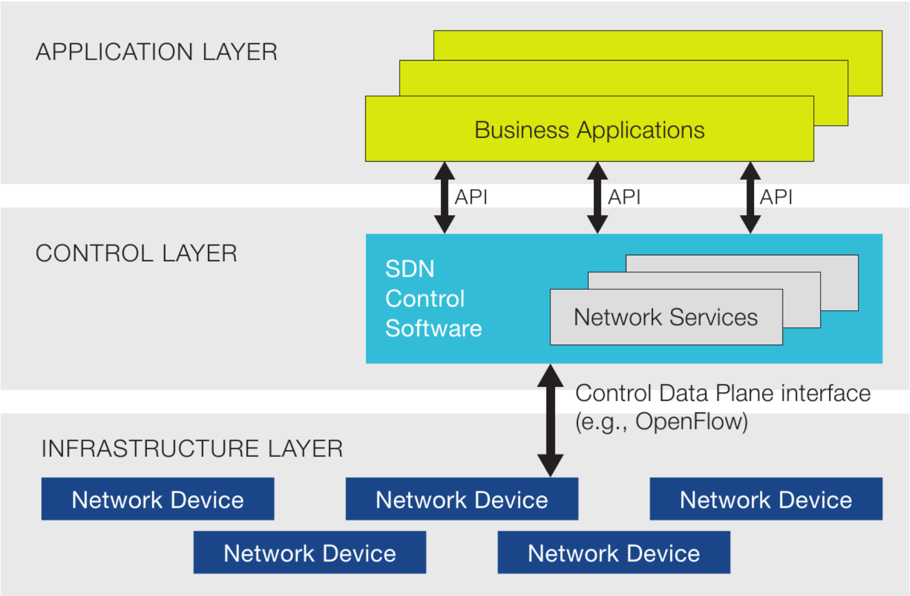


Fig.1: An example SDN architecture (source: https://www.opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf)

Background & Motivations

- There still remain essential challenges of implementing SDN such as the scalability and security of the control plane

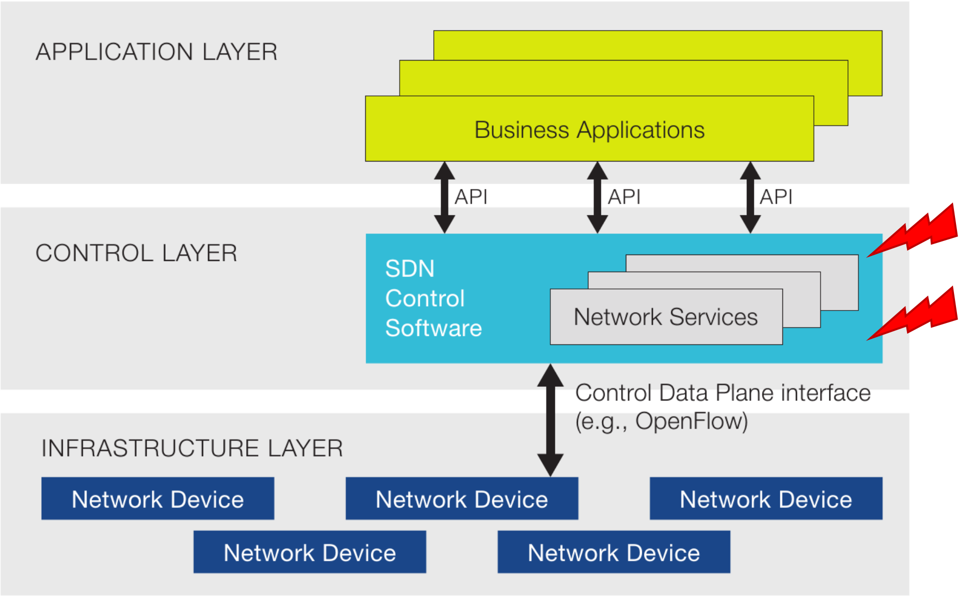


Fig.2: An example SDN architecture with attacks on SDN controller.

Background & Motivations

- We develop a blockchain-enabled multi-controller SDN-based resilient networking management mechanism.

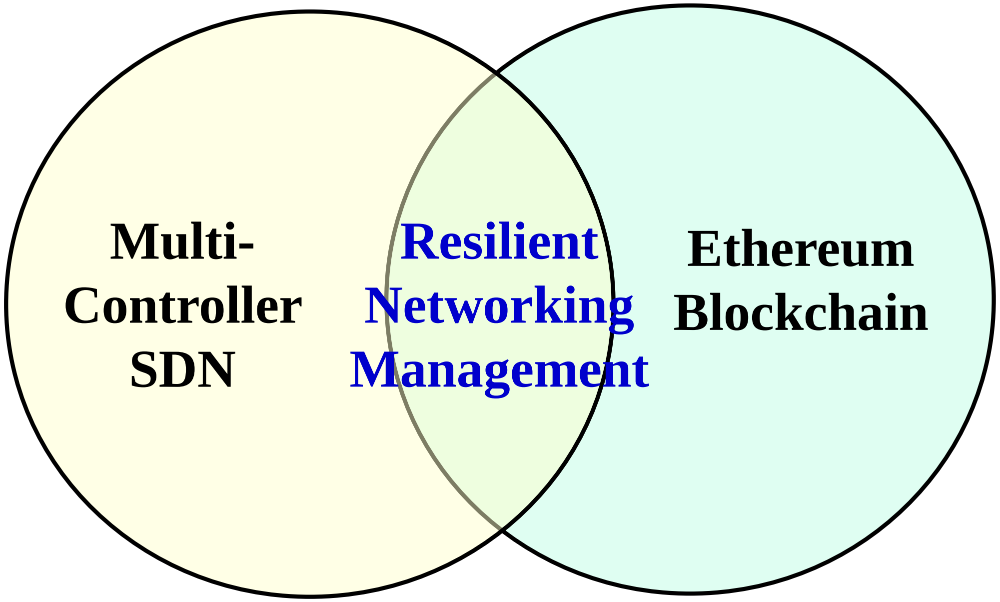


Fig.3: The overview of our proposed mechanism

Background & Motivations

Structure of Creating and Adding New Blocks

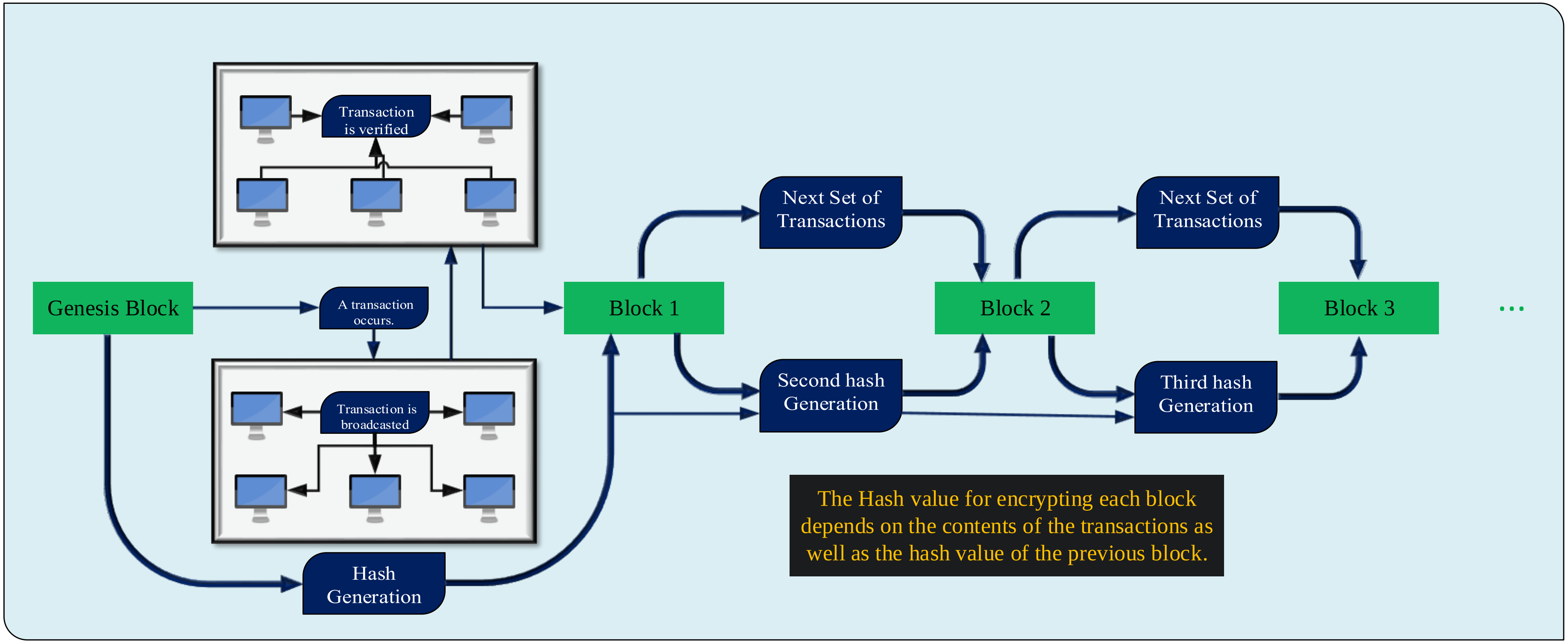


Fig.4: Illustration for blockchain technology.

Our Proposed Networking Solution with Case Studies

- When considering a mission which spans across multiple geographical regions, we are developing a blockchain-powered multi-controller SDN-based networking architecture.

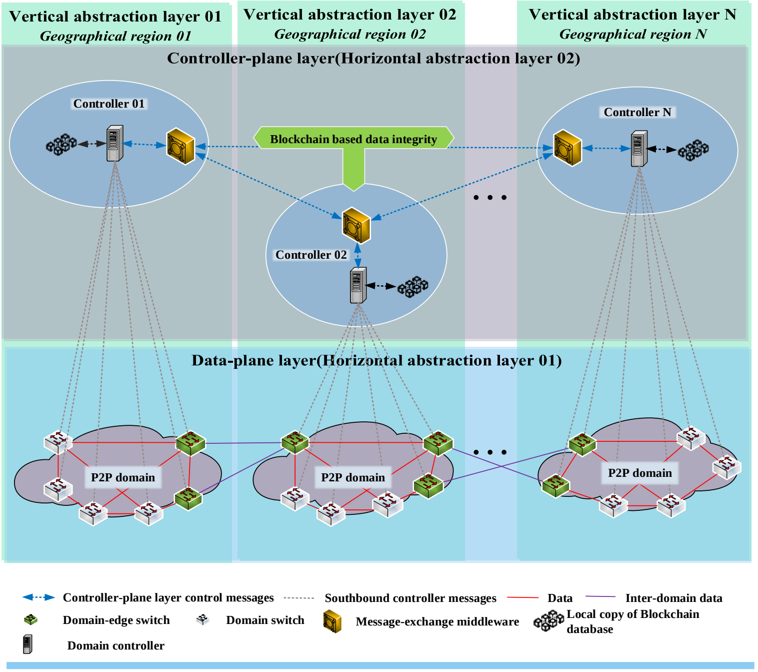


Fig.5: High-level overview of our blockchain-powered multi-controller SDN-based networking interface.

- There will be two horizontal abstraction layers in this architecture. The bottom layer will be the data-plane layer while the top layer will be the controller-plane layer.
- Vertical abstraction layer is divided into several clusters where each cluster is controlled by a dedicated controller.

Our Proposed Networking Solution with Case Studies

Integrity Verification Mechanism

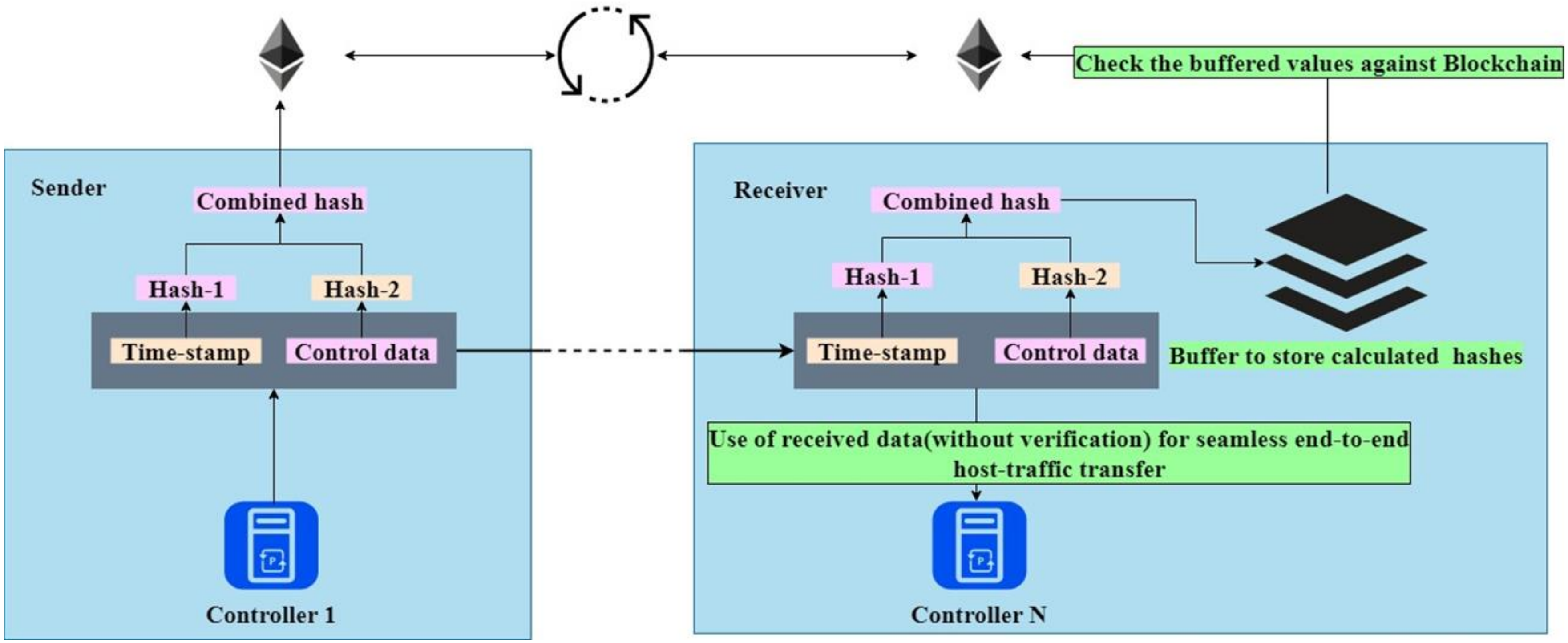


Fig.6: Illustration of securing the integrity of controller-controller communications.

Integrity Verification Mechanism

Case Studies

- We implemented a simple multi-controller topology in which:
  - ➢ There are three Controllers C0, C1 and C2 that can communicate with each other via exchanging ARP messages.
  - ➢ The three associated SDN domains are connected with each other via domain-edge Switches s5, s10 and s11 that are used for inter-domain traffic.

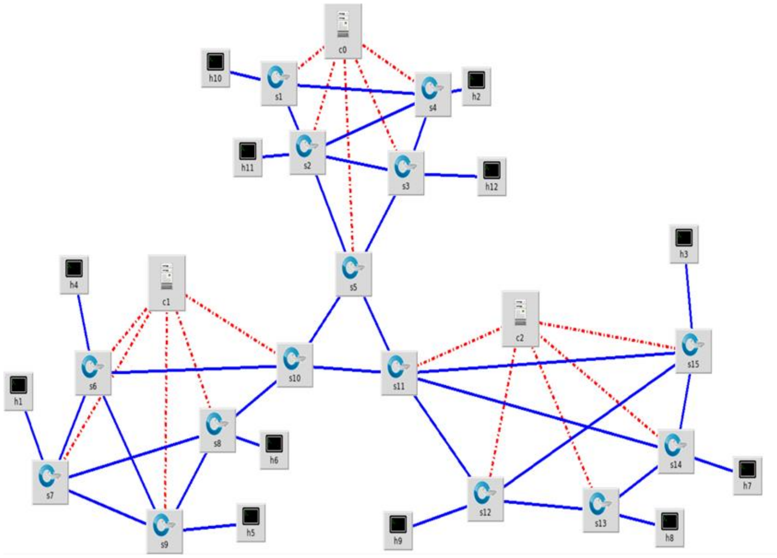


Fig.7: Network topology.

- ➢ The integrity of ARP requests and responses received by a controller is guaranteed by leveraging the functionality of Ethereum blockchain.

Case Studies

- Verifying the validity by pinging from known host to a known destination by a ping from Host 1 at C1 to Host 2 at C0.
  - ➢ We assume that neither the C1 nor C0 knows about the MAC address of each other.
  - ➢ Initially, C1 floods an ARP request, requesting the MAC address of C0 throughout all the domains, where the request will be propagated to C0 and C2.

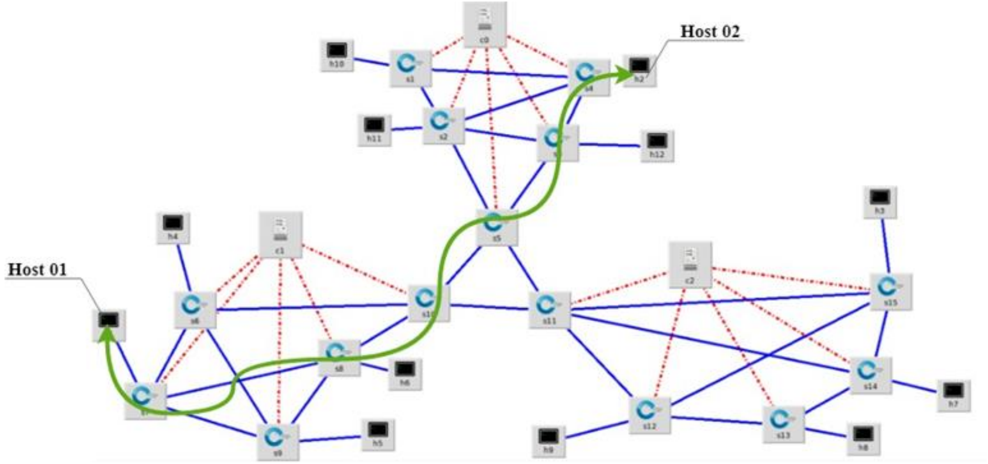


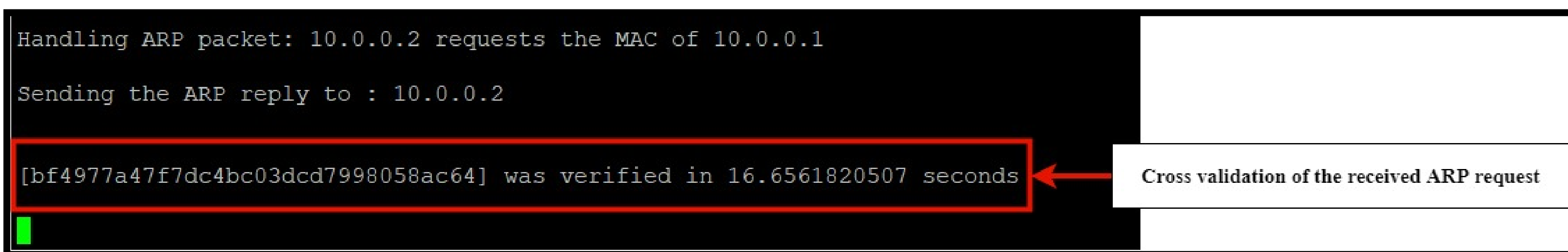
Fig.8: Network topology with datapath.

Case Studies

- Verifying the validity by pinging from known host to a known destination by a ping from Host 1 at C1 to Host 2 at C0.
  - ➢ When C1 sends the ARP request to other controllers, it also hashes the ARP request and uploads it to the blockchain.
  - ➢ The following screenshot shows the corresponding console output of C1.

Case Studies

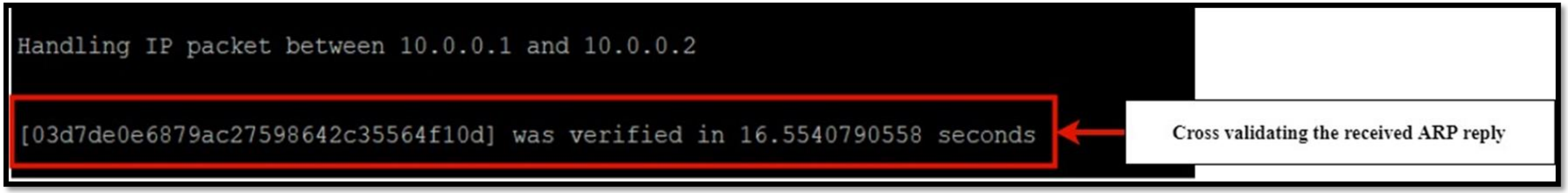
- The ARP request are received by both C0 and C2. Upon reception, they will check the validity of the received ARP requests by hashing them and cross validating the hashes against the blockchain.
- Following console outputs of C0 and C2 show the cross validation of the received ARP request, respectively.



Case Studies

- After the ARP requests have been properly flooded, host 2 at C0 will send the ARP reply to host 1 at C1. After crafting the ARP reply, the controller will upload the hash of the reply to the blockchain.

- Upon the reception of the ARP reply, the destination controller, C1, verifies the hash of the received ARP reply against the blockchain.



Our Proposed Networking Solution with Case Studies

- In our ongoing work, we are exploiting the immutability and the turing completeness of the blockchain to enable network automation.

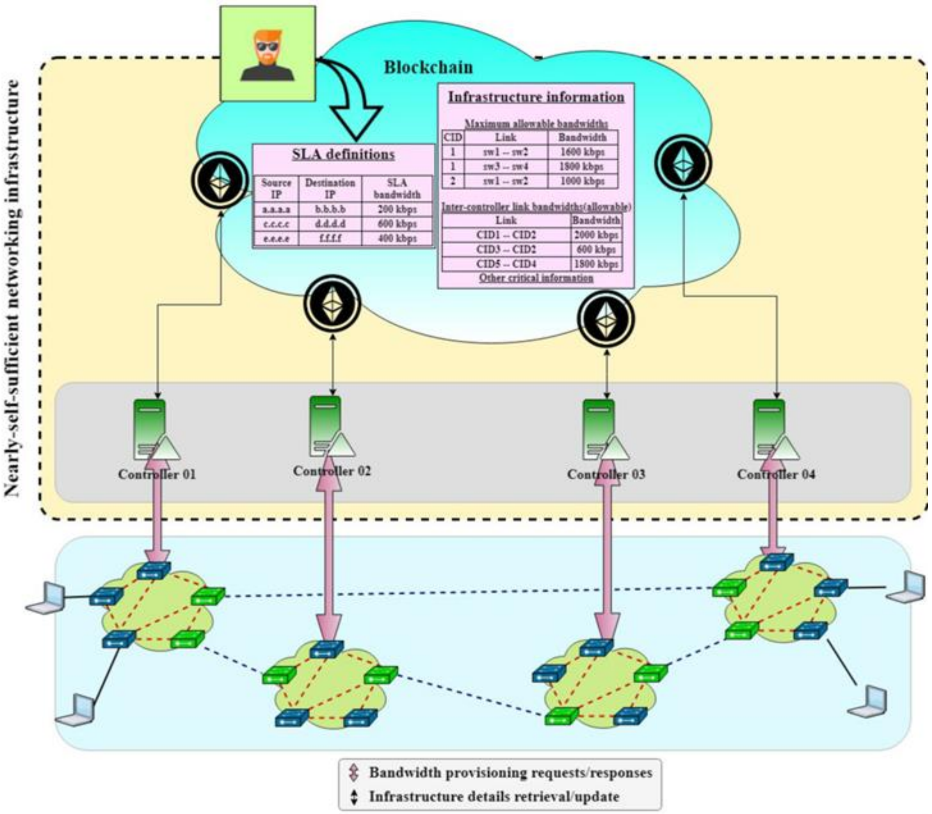


Fig.9: High-level overview of network automation.

- ➢ Autonomous detection of malicious hosts.
  - ✓ To improve security (authorization)
- ➢ Autonomous bandwidth provisioning.
  - ✓ To improve responsiveness and reliability

Autonomous Malicious-Host Detection Mechanism

- The authorization of the host sending ARP requests/replies is verified via blockchain smart contract.
  - ➢ It is assumed that every valid host acquires an IP address using Dynamic Host Configuration Protocol (DHCP) at the SDN controllers.
  - ➢ Once the IP is assigned, the corresponding host's IP-MAC association is recorded in the blockchain platform.

Host IP	MAC address
20.0.0.1	48-2C-6A-1E-59-3D
20.0.0.5	48-2C-6A-1D-5C-ED
20.0.0.9	48-2C-6A-1E-23-4A

  - ➢ The establishment of the end-to-end IP association without a qualified ARP request-reply pair is prevented.

Autonomous Malicious-Host Detection Mechanism

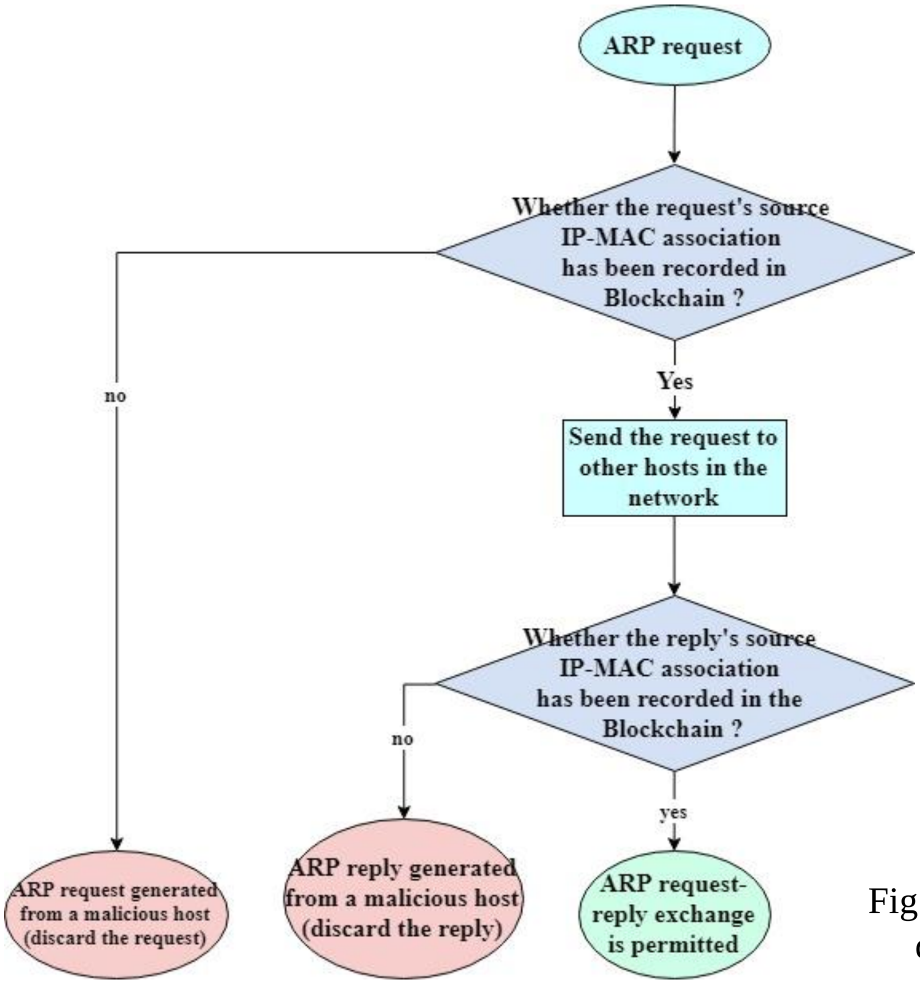
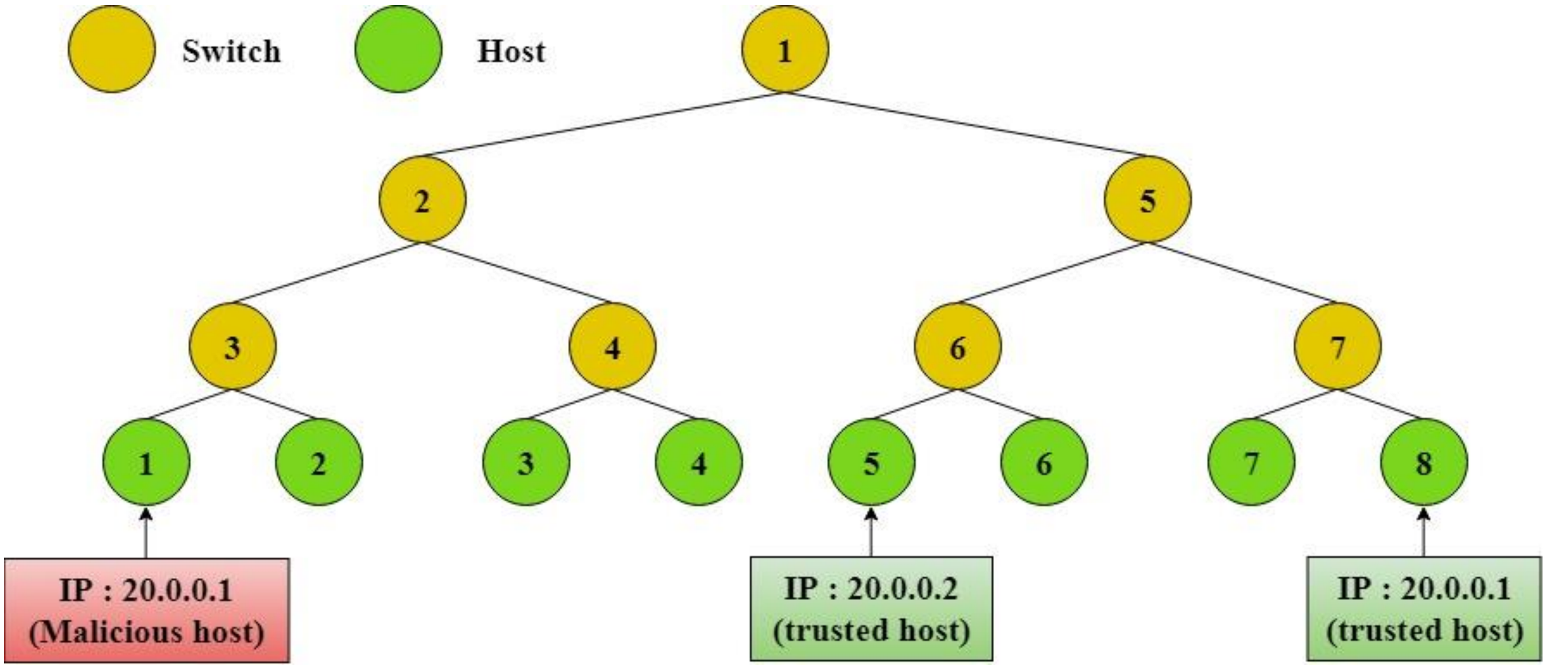


Fig.10: Flow chart for autonomous detection of malicious hosts.

Autonomous Malicious-Host Detection Mechanism

Case Studies

- The SDN network is assumed to have depth = 3 and fan-out = 2.
  - ➢ Initially, all the hosts have the IP addresses of the IPV4 range 20.0.0.0/24.



Autonomous Malicious-Host Detection Mechanism

Case Studies

- Host 8 with IP address 20.0.0.1 and Host 5 with IP address 20.0.0.2 are configured by using DHCP.
- Since both Hosts 5 and 8 get the IPs via DHCP, their corresponding IP-MAC associations are recorded in the blockchain.
  - These two hosts are identified as trusted hosts.

```
Host 20.0.0.2 has a valid MAC address association
Handling ARP packet: 20.0.0.2 requests the MAC of 20.0.0.1
Flooding initial ARP request on all switch edges
```

The authorization of Host 5 sending ARP request is verified at the controller via smart contract.

Autonomous Malicious-Host Detection Mechanism

Case Studies

- Host 8 with IP address 20.0.0.1 and Host 5 with IP address 20.0.0.2 are configured by using DHCP.
- Since both Hosts 5 and 8 get the IPs via DHCP, their corresponding IP-MAC associations are recorded in the blockchain
  - ➢ These two hosts are identified as trusted hosts.

```
Host 20.0.0.1 has a valid MAC address association
Handling ARP reply: 20.0.0.1 responds to 20.0.0.2
Sending the ARP reply to : 20.0.0.2

Host 20.0.0.1 has a invalid MAC address association...ARP reply will be discarded
Handling IP packet between 20.0.0.2 and 20.0.0.1
```

→ The authorization of Host 8 sending ARP reply is verified via smart contract.

→ Malicious host (Host 1) sending ARP reply is detected at the controller. The associated ARP reply is discarded.

Autonomous Bandwidth Provisioning Mechanism

- Blockchain is used here to implement the following.
  - ✓ To impose SLAs (Service Level Agreements) into the networking logic
  - ✓ To keep track of available bandwidths of each links
- The metering and queue functionality of the OpenFlow Switch Specification has been used.
- A certain traffic flow can be categorized into two categories.
  - ✓ Guaranteed traffic(flag = 1)
  - ✓ Best-effort traffic(flag = 0)

Source IP	Destination IP	SLA BW	Flag
a.a.a.a	b.b.b.b	1Mbps	1
c.c.c.c	d.d.d.d	2Mbps	1
e.e.e.e	f.f.f.f	N/A	0

Autonomous Bandwidth Provisioning Mechanism

- Three critical data-structures are defined via the blockchain smart contract.
  - ➢ Inter-controller link bandwidth matrix
    - ✓ To record all the links connecting controllers and their associated bandwidths

Controller pair	Bandwidth
(Controller 01 , Controller 02)	2000 kbps
(Controller 02 , Controller 03)	1800 kbps

  - ➢ Controller link-bandwidth matrix
    - ✓ To record the link bandwidths of the individual controllers

Controller ID	link	Bandwidth
01	(2,3)	2000 kbps
02	(6,8)	400 kbps

Autonomous Bandwidth Provisioning Mechanism

- Three critical data-structures are defined via the blockchain smart contract.
  - Traversal edge-switch matrix
    - ✓ To record the corresponding edge-switch when traversing from one controller to another.

Traversing controllers	Edge-switch datapath-ID (DPID)
(Controller 01 , Controller 02)	01
(Controller 02 , Controller 01)	02
(Controller 02 , Controller 03)	03

Autonomous Bandwidth Provisioning Mechanism

- *Inter-controller link bandwidth matrix* and *controller link bandwidth matrix* are used to
  - Track of all the unallocated bandwidths throughout the network
  - Make decision on the bandwidth provisioning for a given source and destination.
- *Traversal edge-switch matrix* is used to
  - Find the two domain-edge switches of the domain of each controller that participates in the packet switching process for a certain source and destination.
  - The result from the above process is fed to individual controller to achieve the shortest path locally.

Autonomous Bandwidth Provisioning Mechanism

Case Studies

- Leveraging blockchain to identify different traffic types.
  - In our mechanism, the type and suitable bandwidth requirements of each flow is identified based on the entries located in the blockchain platform.

Source IP	Destination IP	Service Level Agreement (SLA) BW	Flag
10.0.0.1	10.0.0.5	N/A	0
10.0.0.2	10.0.0.6	N/A	0
10.0.0.3	10.0.0.7	2Mbps	1
10.0.0.4	10.0.0.8	3Mbps	1

Autonomous Bandwidth Provisioning Mechanism

Case Studies

- Leveraging blockchain to identify different traffic types.
  - ➢ In our mechanism, the type and suitable bandwidth requirements of each follow is identified based on the entries located in the blockchain platform.

A flow is identified as a best-effort flow

```
Requesting traffic type of the flow between : 10.0.0.1 <--> 10.0.0.5

Flag : 0 | Defined SLA : 0 Mbps | This is Best-Effort flow...!!!
Handling ARP packet: 10.0.0.3 requests the MAC of 10.0.0.7
Flooding initial ARP request on all switch edges
Handling ARP reply: 10.0.0.7 responds to 10.0.0.3
Sending the ARP reply to : 10.0.0.3
Handling IP packet between 10.0.0.3 and 10.0.0.7

Requesting traffic type of the flow between : 10.0.0.3 <--> 10.0.0.7

Flag : 1 | Defined SLA : 2 Mbps | This is Guaranteed flow...!!!
```

A flow is identified as a guaranteed flow

Autonomous Bandwidth Provisioning Mechanism

Case Studies

- Leveraging blockchain to keep the online track of all the available bandwidths.
  - ➢ When a request comes from a guaranteed flow, via smart contract the controller will automatically
    - ✓ Check all the available bandwidths throughout the shortest path from source to destination
    - ✓ Make a decision on whether a flow can be allocated.

```
Requesting traffic type of the flow between : 10.0.0.3 <--> 10.0.0.7

Flag : 1 | Defined SLA : 2 Mbps | This is Guaranteed flow...!!!

Available bandwidths in the path : [1000, 1000]

Path installation NEGATIVE...!!! | End-to-end bandwidth provisioning failed...!!!
```

SLA is 2 Mbps

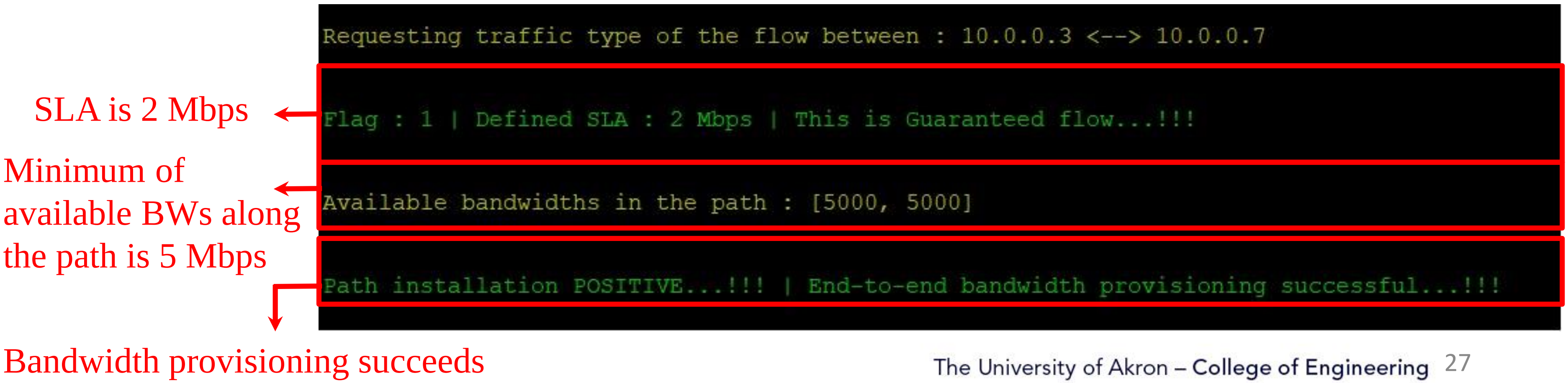
Minimum of available BWs along the path is 1 Mbps

Bandwidth provisioning fails

Autonomous Bandwidth Provisioning Mechanism

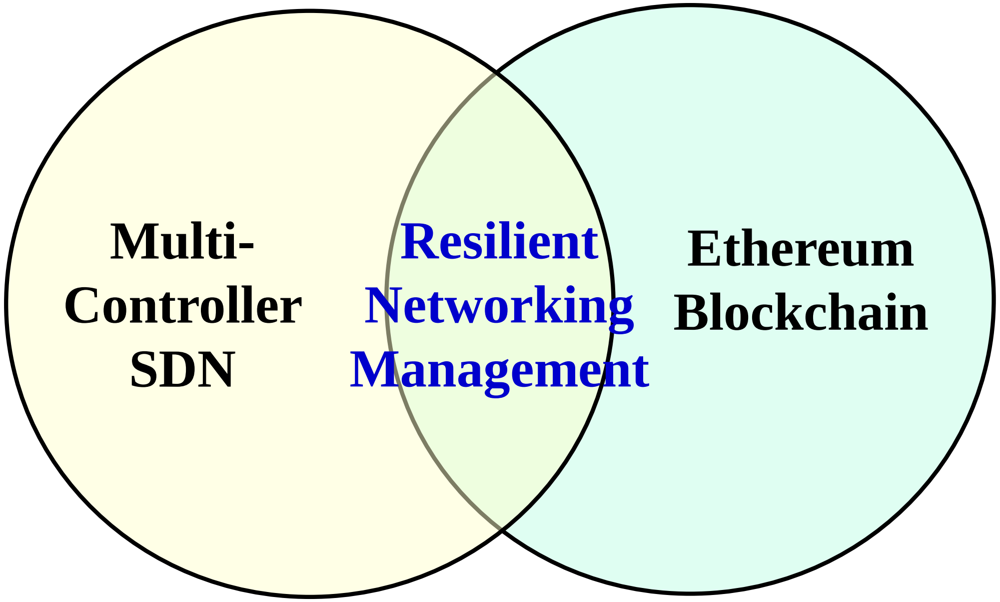
Case Studies

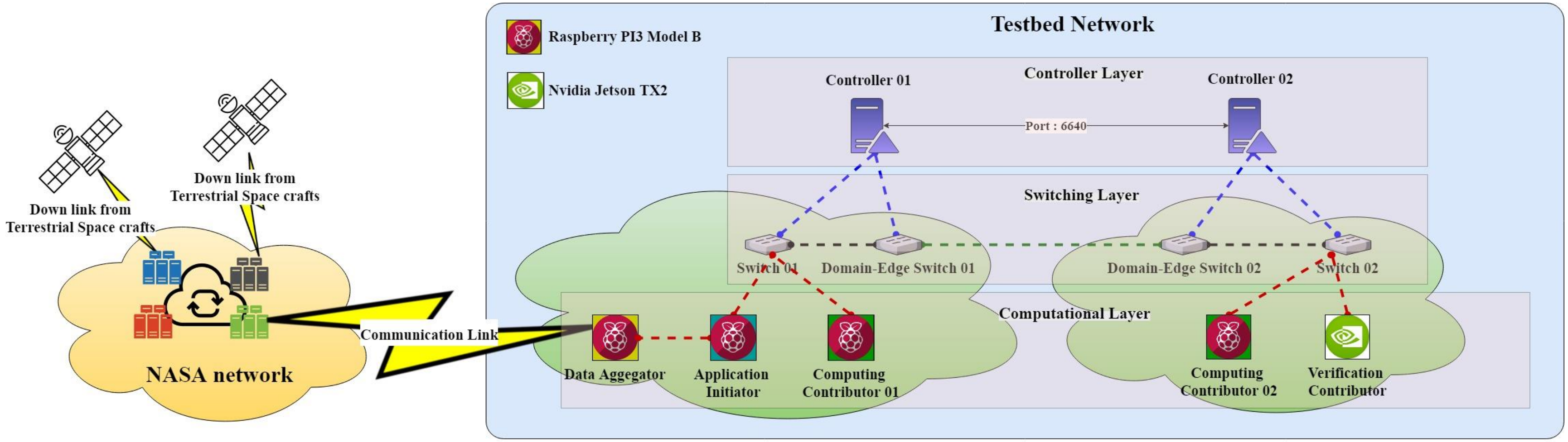
- Leveraging blockchain to keep the online track of all the available bandwidths.
  - ➢ When a request comes from a guaranteed flow, via smart contract the controller will automatically
    - ✓ Check all the available bandwidths throughout the shortest path from source to destination
    - ✓ Make a decision on whether a flow can be allocated.

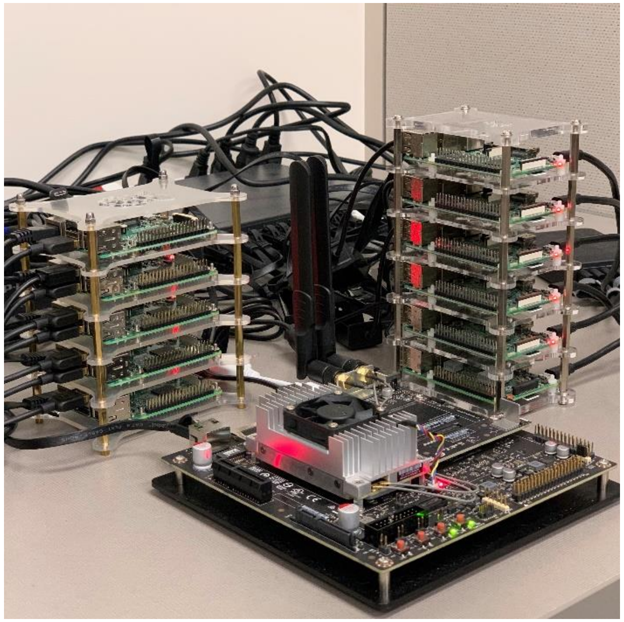


Conclusions

- We develop a blockchain-enabled multi-controller SDN-based resilient networking management mechanism



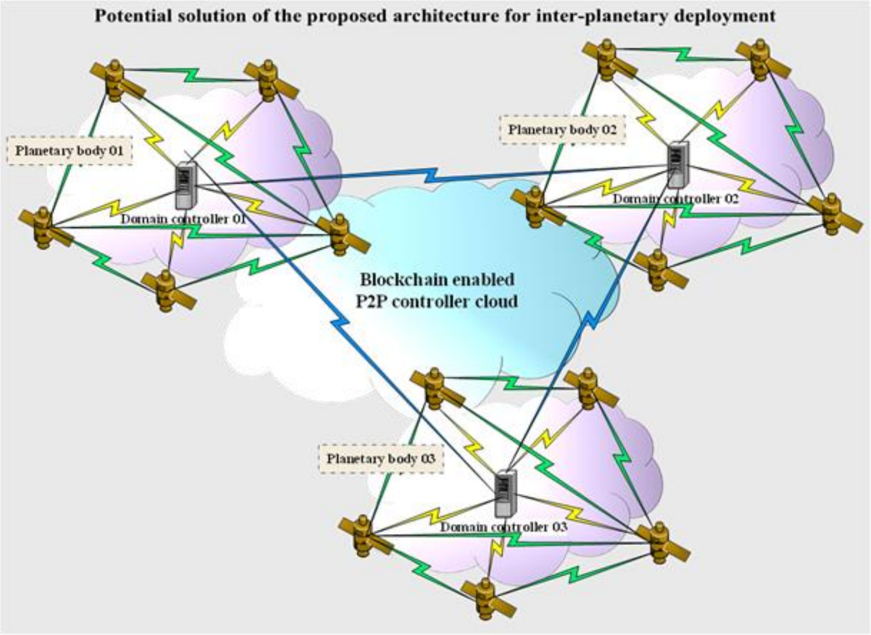




- This is part of the work for the NASA ECF Project "RNCP: A Resilient Networking and Computing Paradigm for NASA Space Exploration".
  - NASA Research Collaborator: Mr. Rigoberto Roche in NASA Glenn Research Center

Conclusions

- Applications
  - Deployment as a means of enabling intelligent and secure deep space communication networks



Conclusions

- Applications
  - Deployment as a means of establishing emergency response networks in a disaster situation

Thank You!
Questions?

The University of Akron
College of Engineering