Why wait? Let’s start computing while data is still on the wire

Shilpi Bhattacharyya,
Dimitrios Katramatos,
Shinjae Yoo

Problem we are trying to solve

With the ever increasing data in transit, storage and early analysis as well as computation is a big challenge.

Why not compute(Analyze, visualize, predict etc.) while data is still in flight? Can we?

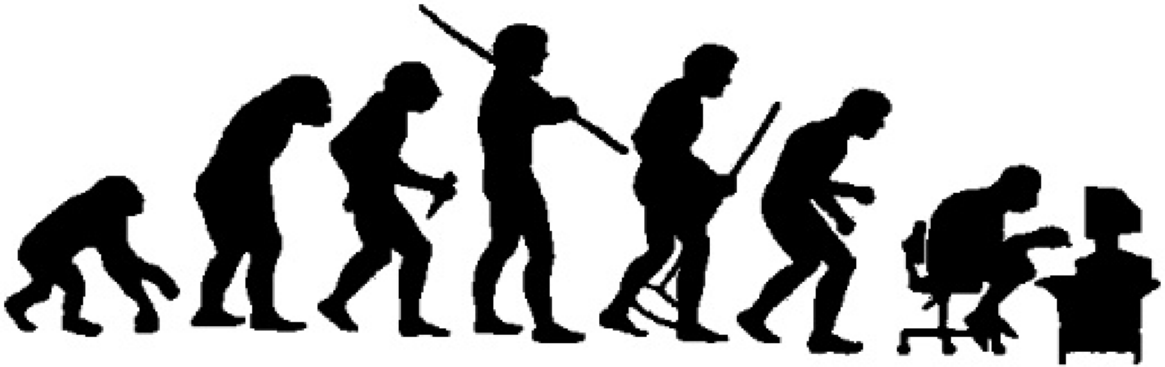
Sure! Let’s attempt to compute in network fabric.

Talk in a nutshell

- “computing in network fabric” - **Analysis on Wire(AoW)** framework.
- Present the **functionality** of AoW with three examples on Forex data, Clickstream data, Solar Sensor data.
- Overall **performance comparison** of the AoW framework in varied scenarios and Analysis.
- **Novelty** of our work from all existing computing paradigms.
- **Limitations** of the AoW Framework.
- **Summary** and **Future work**.

A New Computing Paradigm!!

- Till now we have known to do computations on data at the edge or by redirecting to Cloud.
- We suggest “Computing in the network fabric”.
- The framework designed to solve this problem is named the Analysis on Wire(AoW) framework.



Analysis on Wire(AoW) Framework with Compute-capable Network Nodes

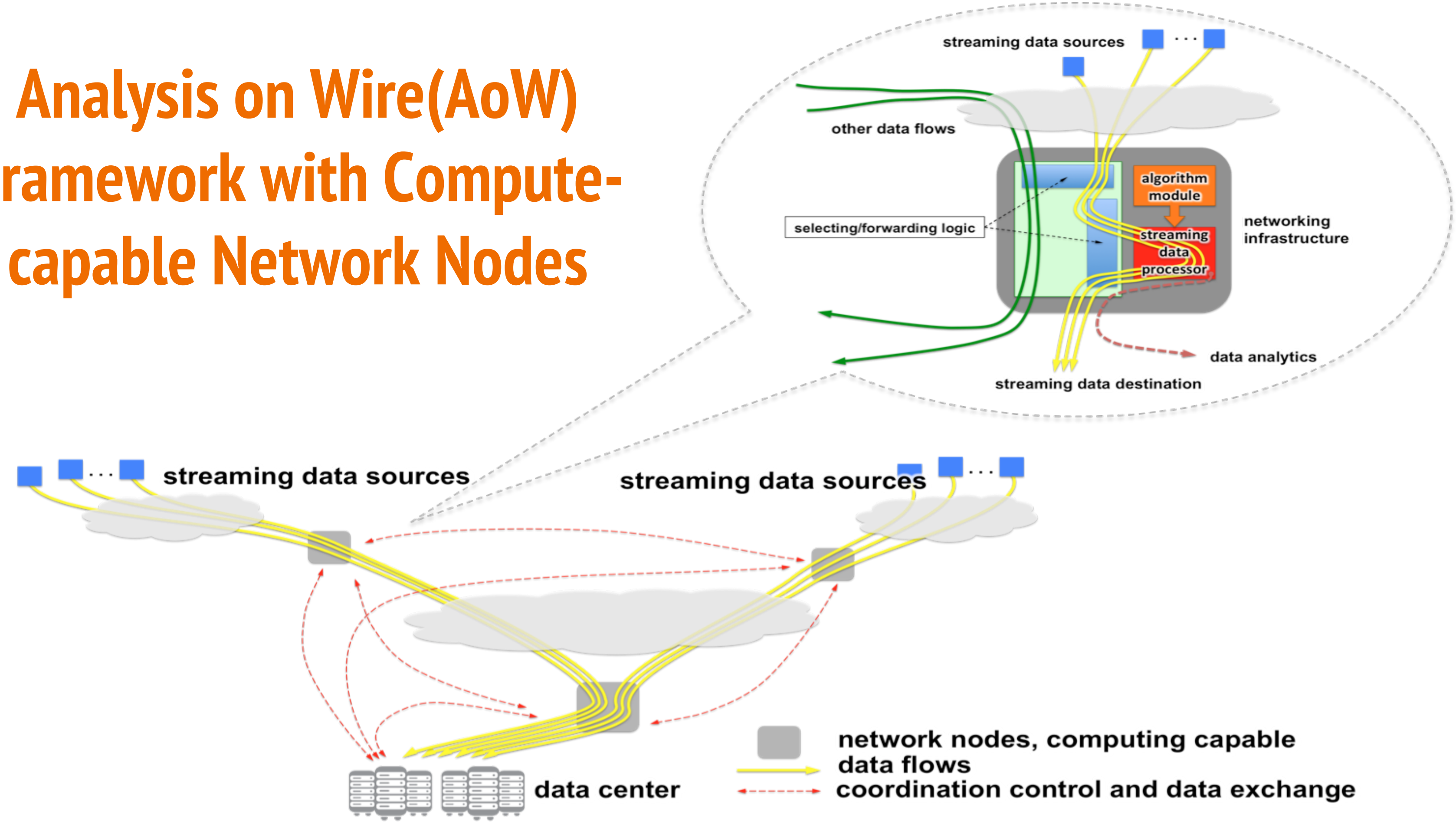


Fig. Analysis on Wire Framework(AoW)

AoW Architecture - Implemented Model

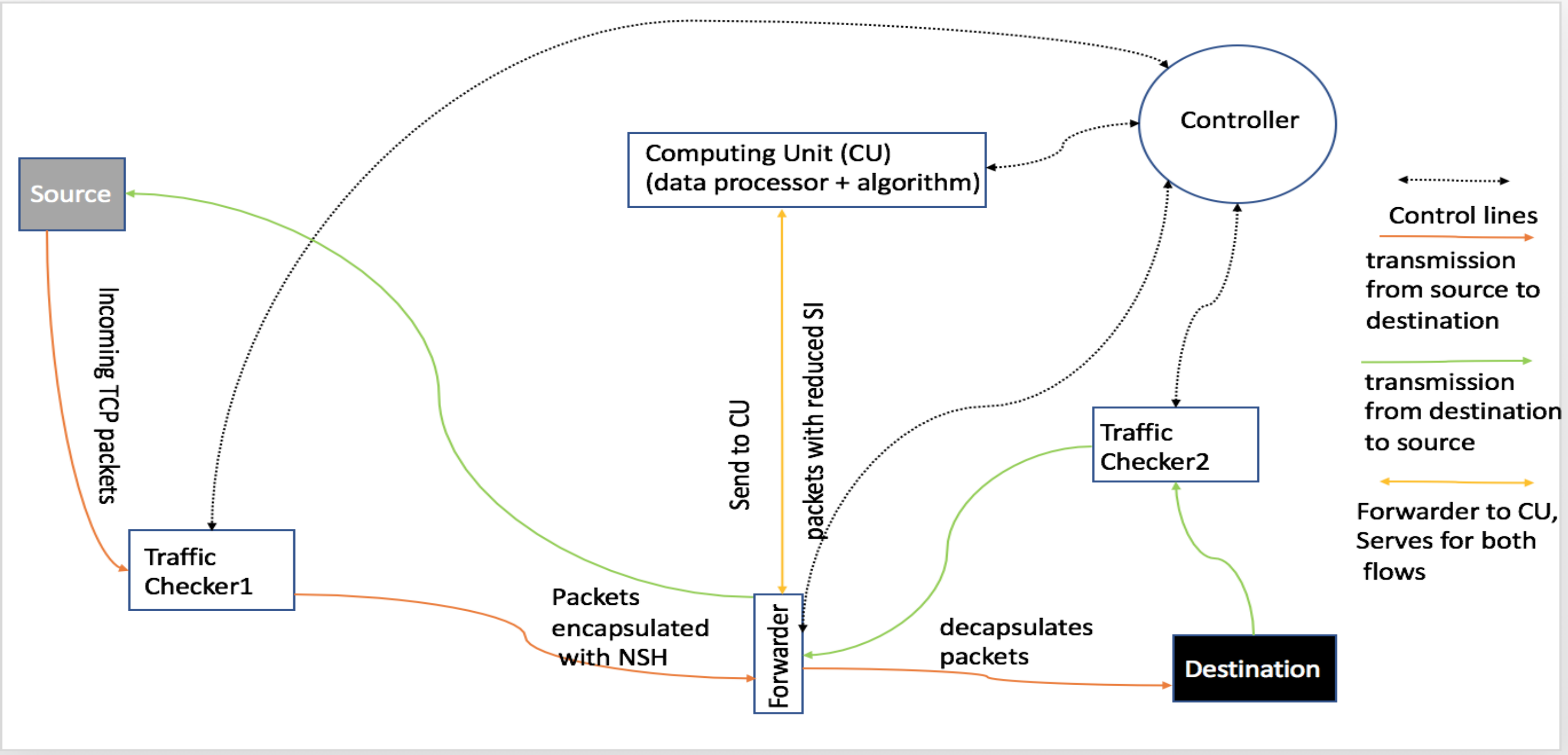
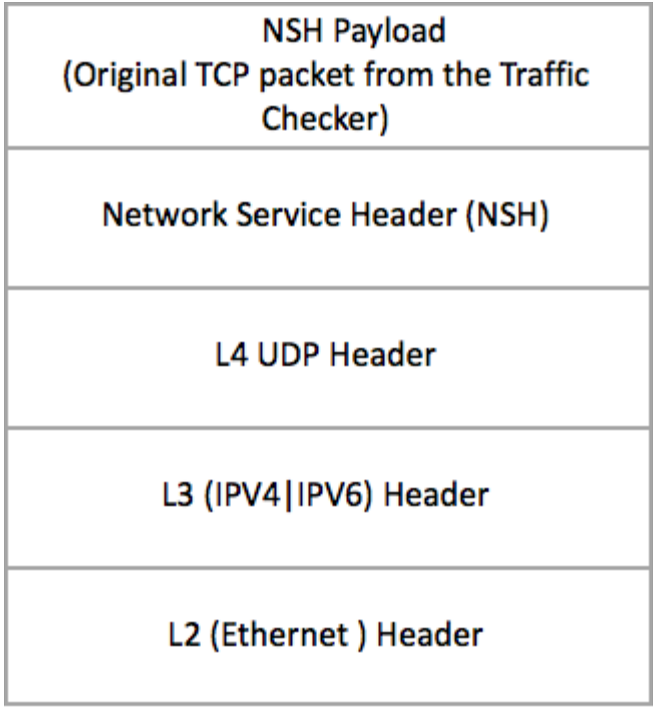

Controller
Computing Unit (CU)
(data processor + algorithm)
Source
Incoming TCP packets
Traffic Checker1
Send to CU
packets with reduced SI
Packets encapsulated with NSH
Forwarder
decapsulates packets
Traffic Checker2
Destination
Control lines
transmission from source to destination
transmission from destination to source
Forwarder to CU, Serves for both flows

Fig. Working model of the Analysis on Wire(AoW) Framework

Use of Network Service Headers in AoW

The traffic checker encapsulates NSH headers into the incoming data packets.

NSH headers are transported through the UDP protocol.



NSH Payload
(Original TCP packet from the Traffic Checker)

Network Service Header (NSH)

L4 UDP Header

L3 (IPV4|IPV6) Header

L2 (Ethernet) Header

Fig. Network Service Header (NSH) UDP Stack

At the end of the chain, the SI(service index) is reduced to zero

Components in the Framework

A Docker-based Service Function Chaining(SFC) architecture with an OpenDayLight Controller.

- Traffic Checker
- Forwarder
- Computing Unit(CU)(Data Processing module + Algorithm module)

Since, we are sending TCP packets, we need a bidirectional framework. Accordingly, we have two traffic checkers, one at source and other at destination. The forwarders with their CU are applicable for both directions.

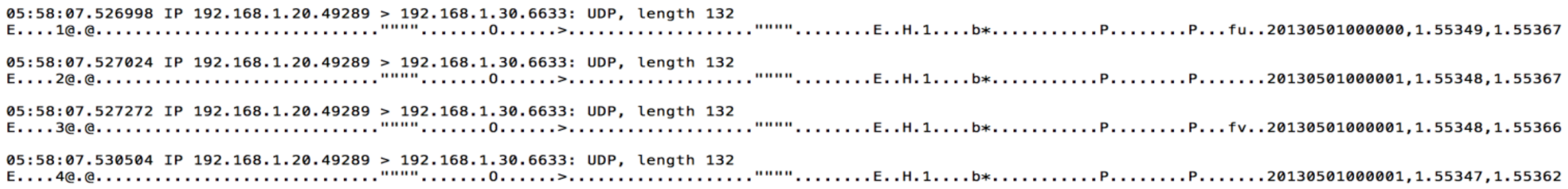
AoW can provide an array of benefits

- Saving resources in Datacenters or Cloud Storage
- Early decision making
- Pre-awareness of impending device failures
- Faster results
- Cybersecurity

Fig. Brookhaven Lab's Scientific Data and Computing Center Group

Examples on the AoW Framework

Visualization and Pattern Recognition on Forex Data



```
05:58:07.526998 IP 192.168.1.20.49289 > 192.168.1.30.6633: UDP, length 132
E....1@.@..............................""""........O.....>.....................""""........E..H.1....b*..........P........P...fu..20130501000000,1.55349,1.55367

05:58:07.527024 IP 192.168.1.20.49289 > 192.168.1.30.6633: UDP, length 132
E....2@.@..............................""""........O.....>.....................""""........E..H.1....b*..........P........P......20130501000001,1.55348,1.55367

05:58:07.527272 IP 192.168.1.20.49289 > 192.168.1.30.6633: UDP, length 132
E....3@.@..............................""""........O.....>.....................""""........E..H.1....b*..........P........P...fv..20130501000001,1.55348,1.55366

05:58:07.530504 IP 192.168.1.20.49289 > 192.168.1.30.6633: UDP, length 132
E....4@.@..............................""""........O.....>.....................""""........E..H.1....b*..........P........P......20130501000001,1.55347,1.55362
```

Fig. Forex Data during an intermediate processing at CU

The green line indicates the ask prices and the blue line represents the bid prices.

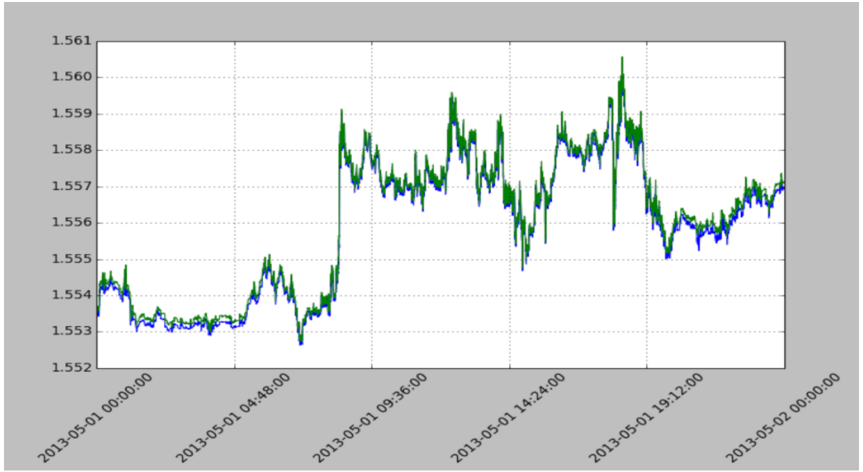
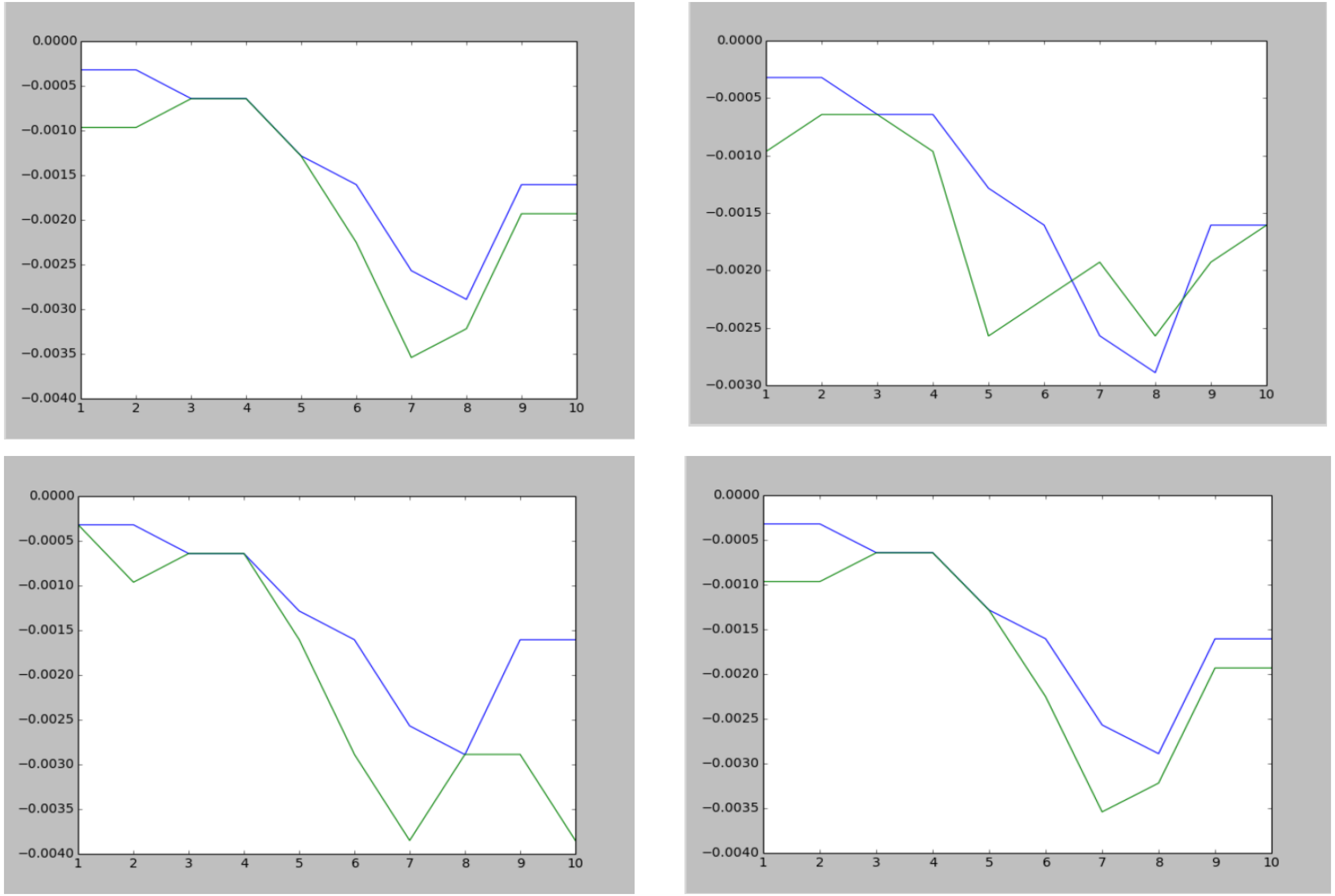


Fig. Forex data visualization

Patterns with greater than 70% similarity



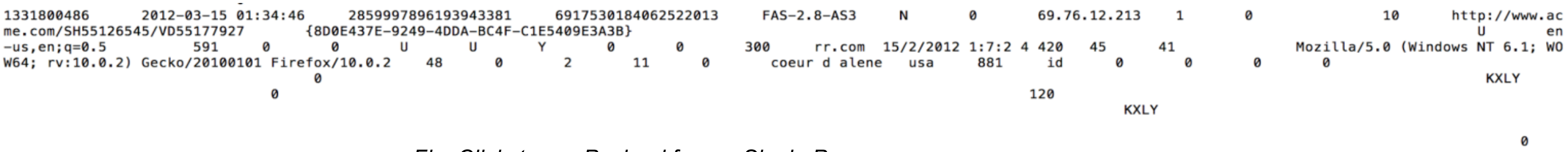
The blue line is the current pattern in question and the green line is the matching similar pattern.

%change = (currPt - startPt)*100/startPt

sim_i = 100.00 - abs(%change(eachPattern[i], patForRecog[i]))

Similarity = $(\sum_i \text{sim}_i)/\sum i$

Clickstream Analysis by Media Publishers



```
1331800486      2012-03-15 01:34:46     2859997896193943381     6917530184062522013     FAS-2.8-AS3     N       0       69.76.12.213    1       0              10      http://www.ac
me.com/SH55126545/VD55177927       {8D0E437E-9249-4DDA-BC4F-C1E5409E3A3B}                                                                                                   U       en
-us,en;q=0.5          591     0       0       U       U       Y       0       0       300      rr.com  15/2/2012 1:7:2 4 420   45      41              Mozilla/5.0 (Windows NT 6.1; WO
W64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2    48      0       2       11      0       coeur d alene   usa     881     id      0       0       0       0
                                    0                                                                                                                                        KXLY
                               0                                                                                        120
                                                                                                                                   KXLY

                                                                                                                                                                                 0
```

Fig. Clickstream Payload from a Single Page

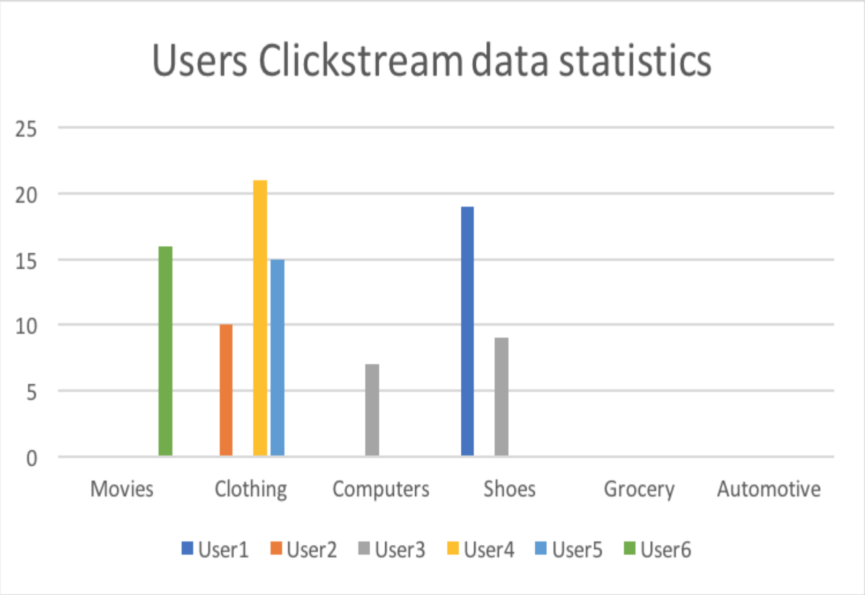
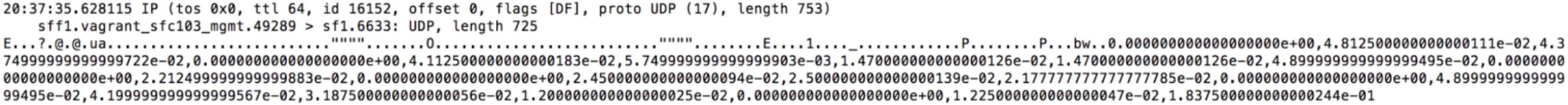


Fig. User Clickstream Data Statistics

Solar Sensors Streaming Data Analysis



```
20:37:35.628115 IP (tos 0x0, ttl 64, id 16152, offset 0, flags [DF], proto UDP (17), length 753)
    sff1.vagrant_sfc103_mgmt.49289 > sf1.6633: UDP, length 725
E...?.@.@.ua.........................""""".......O.........................""""".........E....1...._............P........P...bw..0.0000000000000000000e+00,4.812500000000000111e-02,4.3
74999999999999722e-02,0.0000000000000000000e+00,4.112500000000000183e-02,5.749999999999999903e-03,1.470000000000000126e-02,1.470000000000000126e-02,4.899999999999999495e-02,0.0000000
00000000000e+00,2.212499999999999883e-02,0.0000000000000000000e+00,2.450000000000000094e-02,2.500000000000000139e-02,2.177777777777777785e-02,0.0000000000000000000e+00,4.8999999999999
99495e-02,4.199999999999999567e-02,3.187500000000000056e-02,1.200000000000000025e-02,0.0000000000000000000e+00,1.225000000000000047e-02,1.837500000000000244e-01
```

Fig. Solar Sensor Payload at the Data Processing Unit of CU

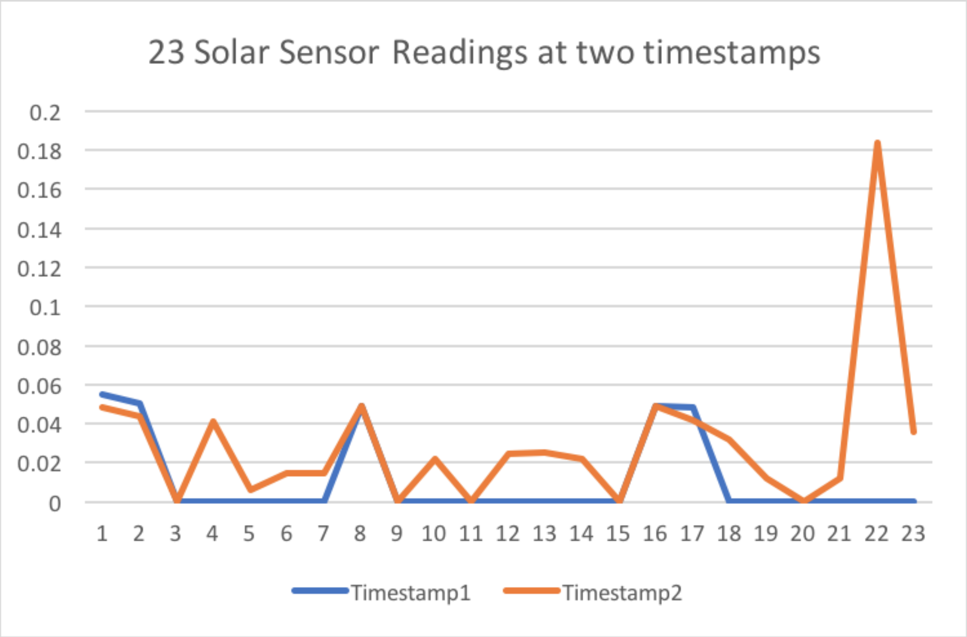
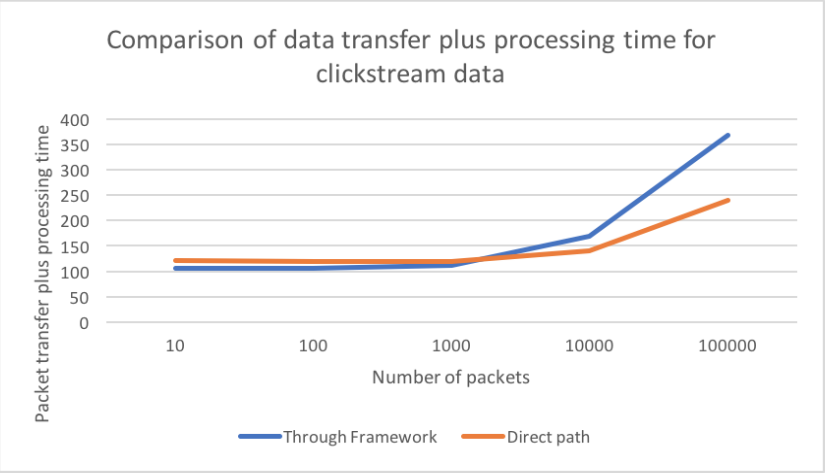
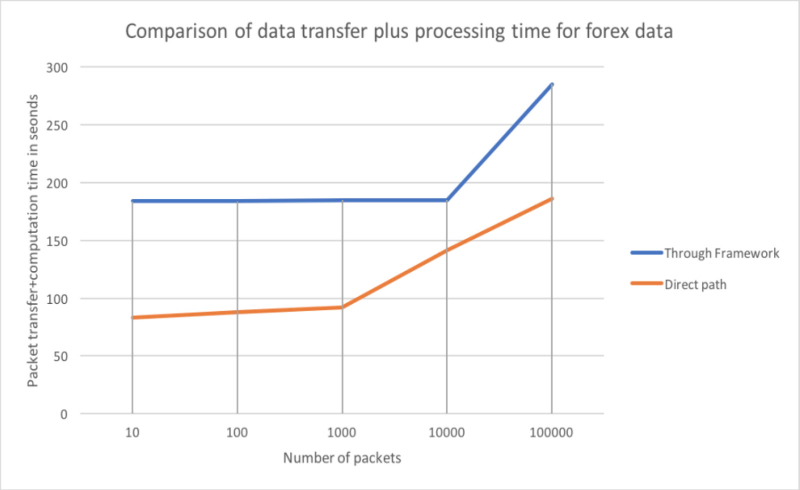
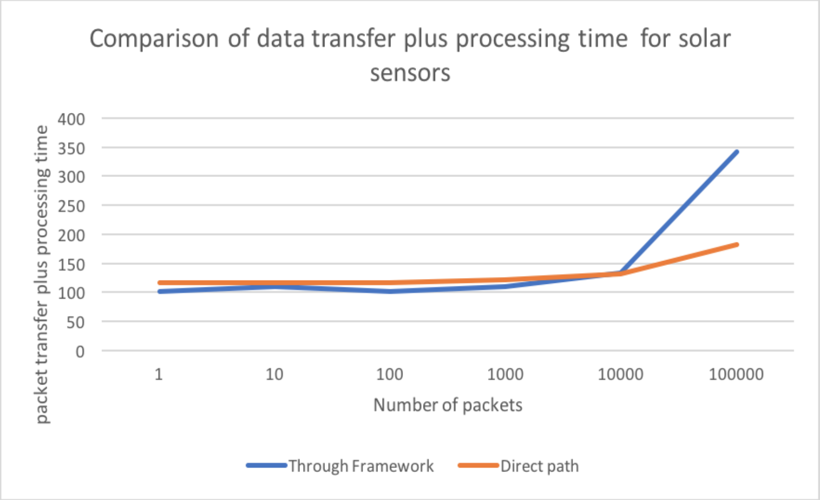


Fig. 23 Solar Sensor Readings at two timestamps

Performance Comparison in Chained and Unchained Path

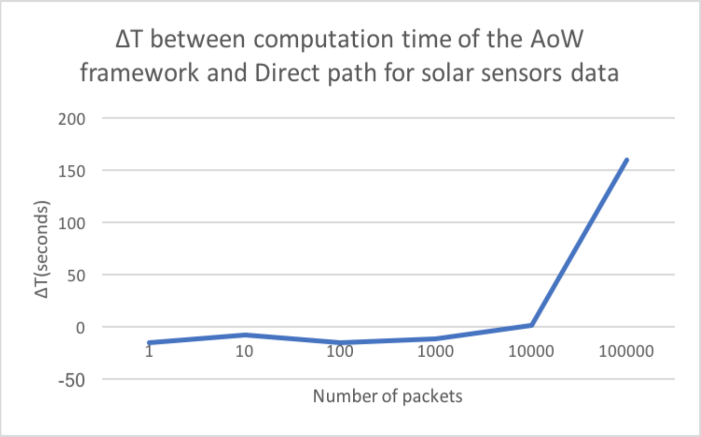


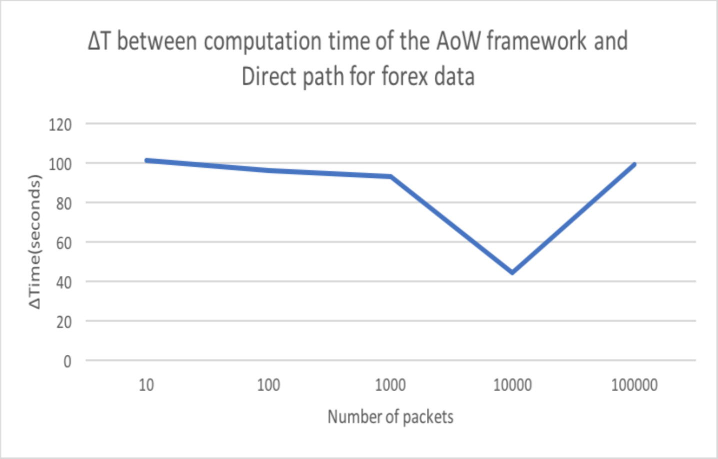


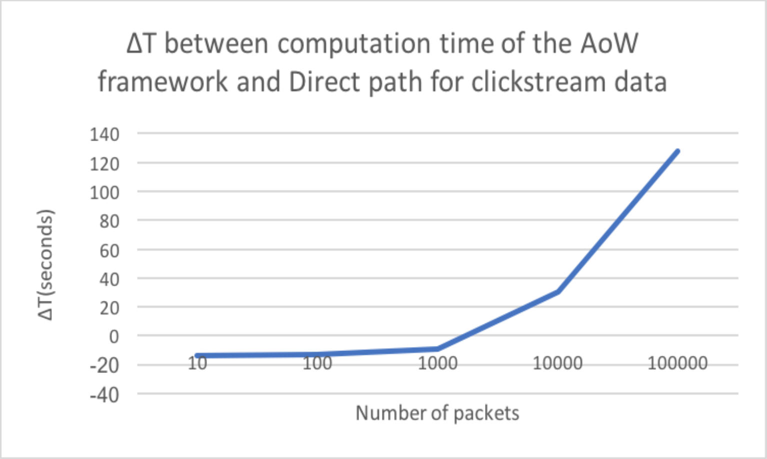


Computation time difference between AoW and Direct path

We denote **Δ**T here as the total time difference experienced by computation on streaming data packets through the framework vs directly to the destination and computation thereafter. The minimum overhead is 44 seconds in case of Forex data at around 60 packets every second. So, the setup does not seem ideal for this kind of computation unless some acceleration is applied at the computing unit or for NSH encapsulation and associated operations.







What about the Prevailing Technologies and Related Work?

- Cloud Computing - storage is performed in virtual data centers that are put together dynamically. Example - Amazon AWS.
- Edge Computing - facilitates the operation of compute, storage, and networking services between end devices and data centers.
- Mist Computing - latest paradigm involving computing in the very end devices found at the edge of the network to assist in the motion of data towards the fog and the cloud.

All paradigms are intimately linked with the Internet of Things (IoT)

Limitations of the AoW Framework

- Operates in a **Congestion-controlled** environment.
- Guarantee of no data loss but, since underlying packet in NSH encapsulation is TCP, **if lost**, sender might repeat sending creating more congestion.
- Not so efficient for **multi-step** huge computations - may need to use multiple nodes or some acceleration.

Summary and Future Work

- **Tremendous** opportunity to perform computations on wire.
- If data rate exceeds the capacity of a basic AoW framework - utilize multiple parallel computing units with a **divide and conquer** approach.
- Independent Data Packets - **defragmentation** on the packets to do complex computing.
- **Acceleration** of computation on the AoW framework by employing **GPUs and FPGAs.**

We simply want to go far beyond the bounds of networking and virtual network functions and devise a framework that can execute any reasonable algorithm on streaming data, while also investigating the behavior and performance of such algorithms to determine the feasibility of solving certain problems on the wire.

Thank You

Questions?