

# An SDN Solution For a GEO and LEO Satellite Transceiver Based on GNU Radio Companion

Ahmed Bannour, Veronique Moeyaert UMONS, Faculty of Engineering  
Electromagnetism and Telecommunications Department, MONS, Belgium  
Email: {ahmed.bannour, veronique.moeyaert}@umons.ac.be

**Abstract**—Nowadays a novel satellite communication solution has to be introduced for the coming satellite networks. Software Defined Networking (SDN) and Software Defined Radio (SDR), can play a key role to innovate and improve satellites-terrestrial communications. GNU radio as SDR and Mininet as SDN, offer the ability to manage the network satellite resources. Through scenario and use cases, this paper provides an in-depth analysis for an effective combination of SDN, GEO, and LEO satellites using GNU Radio simulation and Python3 scripting. This new architecture can bring satellite communications towards a new proposal of data transmission technology to delimit the various potential areas of improvement. That can be pursued by the added value of SDN-GNU Radio technologies in the field of the satellite ground segment. To achieve this result, the newly introduced architecture can be fully controlled at the physical layer level. The solution is validated by the deployment of three scenarios: with SDN, without SDN, and SDN adding a streaming overload. Based on the results of the presented simulation, we found that our architecture can achieve better quality of service through efficient satellite traffic control, when SDN and GNU Radio are deployed.

**Index Terms**—Integration of Satellite and Terrestrial Networks, Next Generation Networks, Software Defined Networking, Space Networking, GNU Radio.

## I. INTRODUCTION

Radio-link availability and services reliability are now highly demanded for the future satellite networks [1], [2].

For example, Low Earth Orbit (LEO) are widely used for Earth observation. The rapid movement of these satellites makes them visible in a given location for a very short period of time, about ten minutes. They must therefore be autonomous to fulfill their mission being rarely in contact with their control center. Another important example is the Geostationary Earth Orbit (GEO). It is by far the most widely used, especially with the development of telecommunications needs, (e.g., it is where live broadcast satellite are located).

Beside their benefits, GEO/LEO satellite configuration and monitoring are yet to be improved to ensure good connectivity and fast configuration. For example, access to configure GEO/LEO satellite is not available unless the satellite is detected by the terrestrial station. Again, the current satellite networking solution is not centralized. Hence robust synchronization and scalability is not very well ensured.

The concept of Software Defined Network (SDN) has dramatically changed the field of modern wireless telecommunications networks. This concept is based on new technologies stepping towards miniaturization and the increase of intelligence of tools and machines. The OSI layered model in the current wireless networks makes the control and the

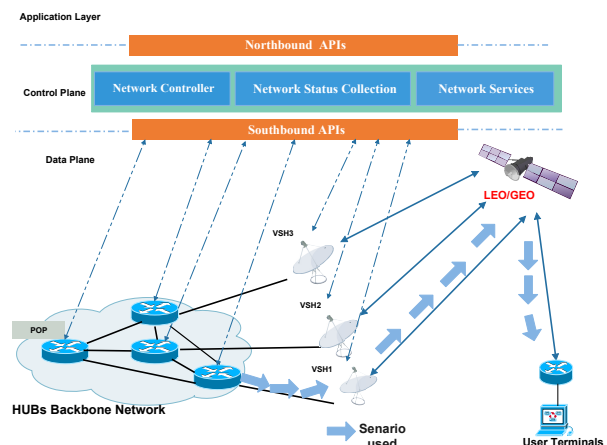


Fig. 1: SDN-GNU Radio experiment scenario for GEO/LEO satellites

management of the infrastructure complex and hard to handle. This is due to the combination of the required data with data responsible for the control. SDN creates an opportunity to solve this problem. Indeed, the global network vision is logically centralized in the control plane and the packet delivery is very efficient in the data plane. Software Defined Network brings flexibility, automation and network customization to networks. Therefore, this type of network also represents an opportunity to facilitate the deployment and management of different wireless networks, things that the previous OSI model cannot offer. SDN has successfully paved the way for next-generation networking, but research work based on wireless networking combined with SDN has recently started.

Henceforth, the combination of Software Defined Network (SDN) and GNU Radio can give a robust and manageable satellite network solution. This can introduce new directives to solve satellite network drawbacks [3]–[5]. The main idea in this paper is to enhance the satellite networking using the new concept of SDN. This new architecture will ensure the decoupling of data, responsible for control from transmission equipment using the controller Python network Operating X-system (POX).

In SDNs, the control plane is placed in a centralized controller that has visibility over the entire network, including the hosts connecting to it. Applications that tell controllers how to program the network using programming interfaces are called "NorthBound" while the Application Programming Interfaces

(APIs) and protocols used by the controller to communicate with network equipment are called "SouthBound" (Fig. 1). The controller POX uses OpenFlow protocol, to administer switches and network equipments.

An OpenFlow network consists of at least one OpenFlow switch and a controller that programs it dynamically. OpenFlow defines a set of criteria for selection on packages required to handle networking operation using Python. These criteria can be non-exhaustive (i.e., the input interface used, the MAC address, the IP address or port number of a packet, both source and destination, or metadata associated with the packet). Next to the criteria for selecting a package, OpenFlow defines actions that can be performed on them. Among possible actions, one can cite:

- Send the packet to an exit gate, a set of gates (group) or to the controller.
- Define the queue in which the packet will be placed in the exit door.
- Modify the packet: header fields, tags, etc ...
- Delete the packet and end its processing.

Logic deterministic packet handling is contained in the switch flow tables. These tables are made up of rules that the controller adds (and removes) dynamically. Each rule includes a set of selection criteria, a priority, and instructions. When a packet arrives at the switch, it is initially assigned empty set of actions. The switch then cycles through the rules in its first table in descending order of priority, looking for a criterion that matches the packet. As soon as the packet is selected by a rule, the cycling is interrupted and the instructions of the rule are executed [6].

A defined satellite networks solution is presented in [7]–[9]. This new OpenSAN architecture gives a fluid control and high efficiency. Operationally Responsive Space (ORS) is introduced by Feng *et al* in [10]. Compared to the latter, this reference paper gives more insights when dealing with emergency response in satellite networking communications. ORS uses OpenFlow protocol in controller to ensure a centralized space to ground network technologies. However the proposed solution does not support the full network visualization.

The related research works, are limited to theoretical analysis, without giving a sufficient realistic SDN satellite networks model to support their concepts. To the best of our knowledge, the combination of GNU radio with SDN to support satellite communications is proposed for the first time in this paper. We can find SDN in some related work enabled in ground satellite portion [11]. However, those solutions degrade network monitoring and reduce flexibility. In this work, we deploy SDN in both ground and space satellite communications portions. This solution will ensure good traffic management and easy configuration. GEO and LEO satellites will be fully controllable in a centralized manner with a flexible global access.

The research work is inspired from the SDR Makerspace (<https://sdrmaker.space/>), which is an initiative of the European Space Agency and the Lieber Space Foundation that brings together researchers from around the world. In fact, the GEO and LEO project uses GNU radio unit that simulates earth-satellite system communications and a variety of aberrations

that a satellite channel can bring. For example, the Carrier Frequency Offset (CFO) due to the Doppler effect (in 435 MHz band), the loss of the variable free space path due to the path of the satellite, or the atmospheric effects (such as atmospheric gases and precipitation) along the flight, have been shown to cause a significant deterioration in the communication quality.

To summarize, we have made the following contributions:

- We enable GEO and LEO satellite communication under SDN architecture using Python3.
- The proposed topology is totally configurable and controllable through POX controller using Openflow protocol.
- A cross layer solution is presented, because transmission is fully configurable from the bit order (physical layer) with GNU radio scripting to logical order (IP traffic) with SDN.

We sectioned the paper as follows. Section II presents the proposed architectures of satellite networks. Then, Section III is devoted to the analysis of each of the scenarios identified. Finally, the main conclusions are elaborated in Section IV

## II. THE PROPOSED ARCHITECTURE

Fig. 1 illustrates an overview of the proposed topology. The architecture is sliced into three planes, the *Data Plane*, the *Control Plane*, and the *Application plane*.

The *Data Plane*. As it was seen in the traditional network principle, the processing resources on satellites should be limited. However, for the SDN network the controller is responsible for policies linked to routing, security, management, and resource allocation through a table flow. In other words, it is the maestro of the entire network. The POX controller then develops policies in relation with GEO and LEO satellites needs and user requirements. These policies are sent to the terminals satellite GEO/LEO to direct traffic between satellites.

The *Control Plane*. It sets up a SDN controller on the GEO/LEO orbit satellites, taking into account the general behavior of the entire topology with respect to network reliability. GEO/LEO satellites collect information about the status of links between satellites and send it to POX controller for the management. During this time, the instructions for network traffic are transmitted to the GEO/LEO satellites. An OpenFlow chart opens, when rules are enabled at satellites level and redirected again to end users.

The *Application Plane*. It presents the programs that define the logic control of the network. The programming interface responsible is called 'Northbound API'. In general the application plane is responsible for ensuring the Quality of Service.

We implement the architecture in a software prototype as shown by the dashed arrow in (blue), with the Southbound API in the Fig. 1. Two nodes are used and each node acts as a router in the topology. Each node must be linked with SDN architecture and the communication between the various physical computers (end user) will be guaranteed through POX controller. Henceforth, the different instances can establish communication between one another and transmit packets via SDN. As illustrated in Fig. 1 the architecture presents two

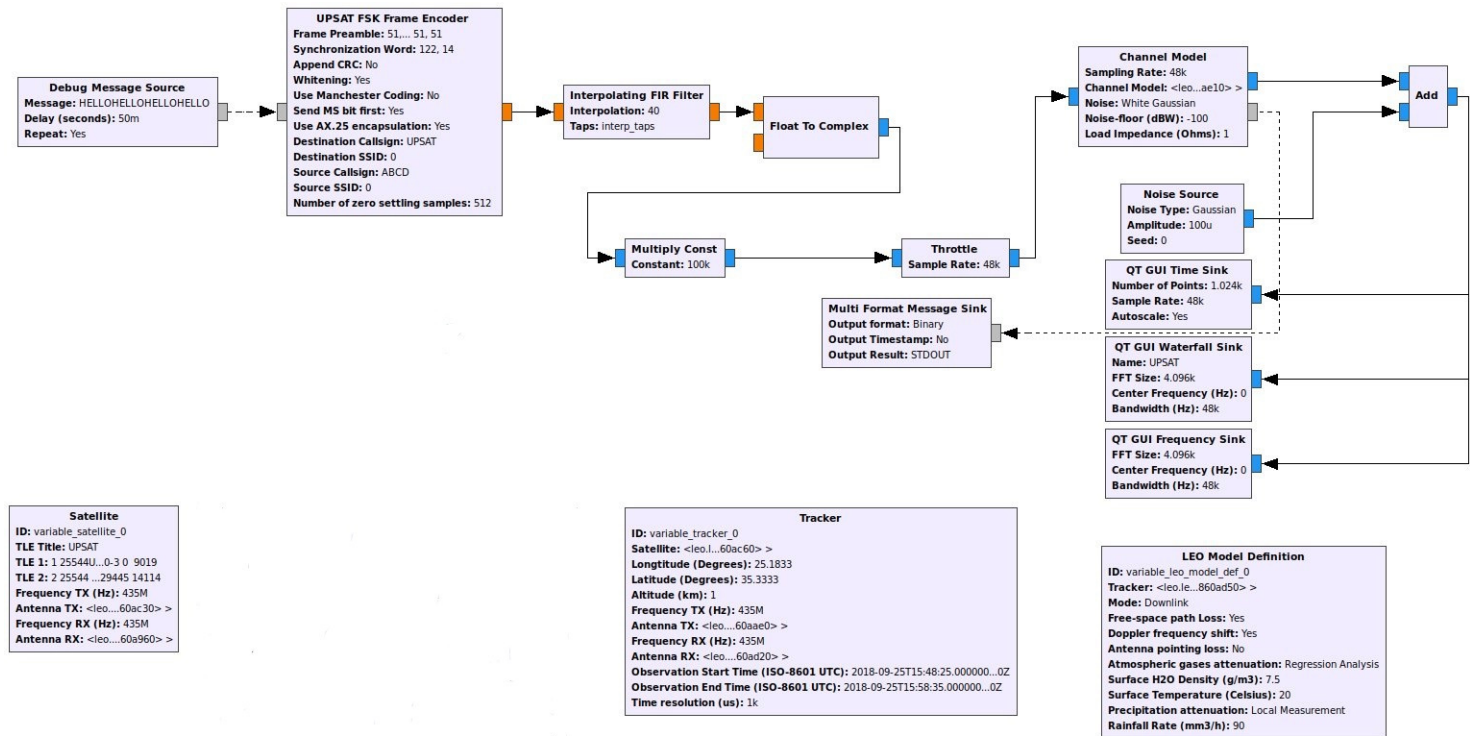


Fig. 2: GEO/LEO satellites transceiver based on GNU Radio Companion

APIs: Southbound API, and Northbound API. Southbound API interface represents the communication interfaces, which allows the SDN controller to interact with the infrastructure layer, such as switches, computers and routers. To maintain good communication with the South interface, OpenFlow protocol was enabled. Northbound APIs are used to program the transmission equipment, using the network abstraction provided by the control plane.

#### The Work Flow

- We start by creating an SDN topology with a Python3 scripting. The first step is to install Mininet (<http://mininet.org/>) to create the proposed architecture. Mininet contains the SDN controller POX which is responsible for the whole management of the network. We have also implemented the POX controller that can support OpenFlow protocol and it is written in Python3.
- Then we implemented a Python3 script, which interacts directly with GNURadio Companion to ensure GEO/LEO satellite transmission chain in the SDN topology as illustrated in Fig. 2.
- Finally we evaluated the performance of the prototype compared to traditional satellite transmission with three scenarios: without SDN, with SDN only and finally SDN with streaming load.

The ground earth station is represented in Fig. 2 blocs, as an instance of the *Tracker* class. It contains parameters for all the information needed to fully describe a ground station that follows a satellite in orbit, such as:

- The coordinates of the ground station (Latitude, Longitude and Altitude)

- TX / RX operating frequencies
- TX / RX antenna objects
- The observation period in UTC ISO-8601 format
- The resolution of the observation time.

A satellite is represented in Fig. 2, as an instance of the class *Satellite*. It contains parameters for all the information required to describe a satellite in orbit, such as:

- An orbit description.
- Satellite operating TX / RX frequencies
- TX / RX antenna objects.

Different types of antennas are supported by Fig. 2, including *Yagi*, *Helix*, *Monopole*, *Dipole* and *Parabolic reflector*. Each is described by a C++ class which extends the parent generic antenna and appropriately implements pure virtual functions. In addition, each class accepts different parameters and can implement various functions, based on the corresponding type. The AMSAT-IARU Link Model spreadsheet is used as a reference for calculating gain, beam-width and cutoff gain etc...

### III. NUMERICAL RESULTS

To showcase the robustness of the proposed architecture and to demonstrate its feasibility, three scenarios are introduced, to emulate the space to ground satellite communication. We built scenes: with SDN only, without SDN, and finally SDN with streaming load. These scenarios help to see in a closer window whether this combination can improve the satellite network in terms of latency, throughput and reliability. Comparison between scenarios can also help to judge the impact of

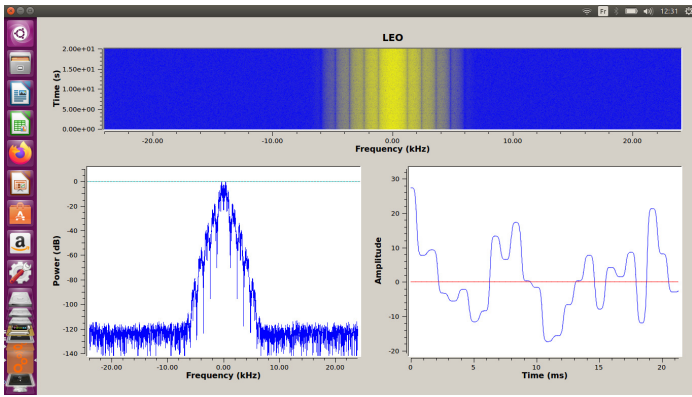


Fig. 3: Example of the developed LEO satellite transceiver using Python3 scripting.

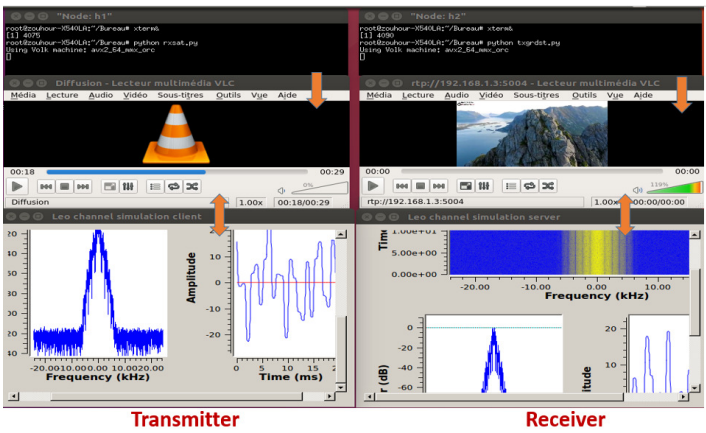


Fig. 5: GEO/LEO satellites transceiver SDN and GNU Radio based, with successful streaming transmission

```

root@zouhour-X540LH:~#ssh root@127.0.0.1:6633
unable to contact the remote controller at 127.0.0.1:6633
connecting to remote controller at 127.0.0.1:6633
*** Adding hosts
*** Adding links
*** Starting network
*** Starting controller
*** Starting i switches
*** Starting CL1
*** Starting CL2
*** Starting CL3
*** Starting CL4
*** Starting CL5
*** Starting CL6
*** Starting CL7
*** Starting CL8
*** Starting CL9
*** Starting CL10
*** Starting CL11
*** Starting CL12
*** Starting CL13
*** Starting CL14
*** Starting CL15
*** Starting CL16
*** Starting CL17
*** Starting CL18
*** Starting CL19
*** Starting CL20
*** Starting CL21
*** Starting CL22
*** Starting CL23
*** Starting CL24
*** Starting CL25
*** Starting CL26
*** Starting CL27
*** Starting CL28
*** Starting CL29
*** Starting CL30
*** Starting CL31
*** Starting CL32
*** Starting CL33
*** Starting CL34
*** Starting CL35
*** Starting CL36
*** Starting CL37
*** Starting CL38
*** Starting CL39
*** Starting CL40
*** Starting CL41
*** Starting CL42
*** Starting CL43
*** Starting CL44
*** Starting CL45
*** Starting CL46
*** Starting CL47
*** Starting CL48
*** Starting CL49
*** Starting CL50
*** Starting CL51
*** Starting CL52
*** Starting CL53
*** Starting CL54
*** Starting CL55
*** Starting CL56
*** Starting CL57
*** Starting CL58
*** Starting CL59
*** Starting CL60
*** Starting CL61
*** Starting CL62
*** Starting CL63
*** Starting CL64
*** Starting CL65
*** Starting CL66
*** Starting CL67
*** Starting CL68
*** Starting CL69
*** Starting CL70
*** Starting CL71
*** Starting CL72
*** Starting CL73
*** Starting CL74
*** Starting CL75
*** Starting CL76
*** Starting CL77
*** Starting CL78
*** Starting CL79
*** Starting CL80
*** Starting CL81
*** Starting CL82
*** Starting CL83
*** Starting CL84
*** Starting CL85
*** Starting CL86
*** Starting CL87
*** Starting CL88
*** Starting CL89
*** Starting CL90
*** Starting CL91
*** Starting CL92
*** Starting CL93
*** Starting CL94
*** Starting CL95
*** Starting CL96
*** Starting CL97
*** Starting CL98
*** Starting CL99
*** Starting CL100
*** Starting CL101
*** Starting CL102
*** Starting CL103
*** Starting CL104
*** Starting CL105
*** Starting CL106
*** Starting CL107
*** Starting CL108
*** Starting CL109
*** Starting CL110
*** Starting CL111
*** Starting CL112
*** Starting CL113
*** Starting CL114
*** Starting CL115
*** Starting CL116
*** Starting CL117
*** Starting CL118
*** Starting CL119
*** Starting CL120
*** Starting CL121
*** Starting CL122
*** Starting CL123
*** Starting CL124
*** Starting CL125
*** Starting CL126
*** Starting CL127
*** Starting CL128
*** Starting CL129
*** Starting CL130
*** Starting CL131
*** Starting CL132
*** Starting CL133
*** Starting CL134
*** Starting CL135
*** Starting CL136
*** Starting CL137
*** Starting CL138
*** Starting CL139
*** Starting CL140
*** Starting CL141
*** Starting CL142
*** Starting CL143
*** Starting CL144
*** Starting CL145
*** Starting CL146
*** Starting CL147
*** Starting CL148
*** Starting CL149
*** Starting CL150
*** Starting CL151
*** Starting CL152
*** Starting CL153
*** Starting CL154
*** Starting CL155
*** Starting CL156
*** Starting CL157
*** Starting CL158
*** Starting CL159
*** Starting CL160
*** Starting CL161
*** Starting CL162
*** Starting CL163
*** Starting CL164
*** Starting CL165
*** Starting CL166
*** Starting CL167
*** Starting CL168
*** Starting CL169
*** Starting CL170
*** Starting CL171
*** Starting CL172
*** Starting CL173
*** Starting CL174
*** Starting CL175
*** Starting CL176
*** Starting CL177
*** Starting CL178
*** Starting CL179
*** Starting CL180
*** Starting CL181
*** Starting CL182
*** Starting CL183
*** Starting CL184
*** Starting CL185
*** Starting CL186
*** Starting CL187
*** Starting CL188
*** Starting CL189
*** Starting CL190
*** Starting CL191
*** Starting CL192
*** Starting CL193
*** Starting CL194
*** Starting CL195
*** Starting CL196
*** Starting CL197
*** Starting CL198
*** Starting CL199
*** Starting CL200
*** Starting CL201
*** Starting CL202
*** Starting CL203
*** Starting CL204
*** Starting CL205
*** Starting CL206
*** Starting CL207
*** Starting CL208
*** Starting CL209
*** Starting CL210
*** Starting CL211
*** Starting CL212
*** Starting CL213
*** Starting CL214
*** Starting CL215
*** Starting CL216
*** Starting CL217
*** Starting CL218
*** Starting CL219
*** Starting CL220
*** Starting CL221
*** Starting CL222
*** Starting CL223
*** Starting CL224
*** Starting CL225
*** Starting CL226
*** Starting CL227
*** Starting CL228
*** Starting CL229
*** Starting CL230
*** Starting CL231
*** Starting CL232
*** Starting CL233
*** Starting CL234
*** Starting CL235
*** Starting CL236
*** Starting CL237
*** Starting CL238
*** Starting CL239
*** Starting CL240
*** Starting CL241
*** Starting CL242
*** Starting CL243
*** Starting CL244
*** Starting CL245
*** Starting CL246
*** Starting CL247
*** Starting CL248
*** Starting CL249
*** Starting CL250
*** Starting CL251
*** Starting CL252
*** Starting CL253
*** Starting CL254
*** Starting CL255
*** Starting CL256
*** Starting CL257
*** Starting CL258
*** Starting CL259
*** Starting CL260
*** Starting CL261
*** Starting CL262
*** Starting CL263
*** Starting CL264
*** Starting CL265
*** Starting CL266
*** Starting CL267
*** Starting CL268
*** Starting CL269
*** Starting CL270
*** Starting CL271
*** Starting CL272
*** Starting CL273
*** Starting CL274
*** Starting CL275
*** Starting CL276
*** Starting CL277
*** Starting CL278
*** Starting CL279
*** Starting CL280
*** Starting CL281
*** Starting CL282
*** Starting CL283
*** Starting CL284
*** Starting CL285
*** Starting CL286
*** Starting CL287
*** Starting CL288
*** Starting CL289
*** Starting CL290
*** Starting CL291
*** Starting CL292
*** Starting CL293
*** Starting CL294
*** Starting CL295
*** Starting CL296
*** Starting CL297
*** Starting CL298
*** Starting CL299
*** Starting CL300
*** Starting CL301
*** Starting CL302
*** Starting CL303
*** Starting CL304
*** Starting CL305
*** Starting CL306
*** Starting CL307
*** Starting CL308
*** Starting CL309
*** Starting CL310
*** Starting CL311
*** Starting CL312
*** Starting CL313
*** Starting CL314
*** Starting CL315
*** Starting CL316
*** Starting CL317
*** Starting CL318
*** Starting CL319
*** Starting CL320
*** Starting CL321
*** Starting CL322
*** Starting CL323
*** Starting CL324
*** Starting CL325
*** Starting CL326
*** Starting CL327
*** Starting CL328
*** Starting CL329
*** Starting CL330
*** Starting CL331
*** Starting CL332
*** Starting CL333
*** Starting CL334
*** Starting CL335
*** Starting CL336
*** Starting CL337
*** Starting CL338
*** Starting CL339
*** Starting CL340
*** Starting CL341
*** Starting CL342
*** Starting CL343
*** Starting CL344
*** Starting CL345
*** Starting CL346
*** Starting CL347
*** Starting CL348
*** Starting CL349
*** Starting CL350
*** Starting CL351
*** Starting CL352
*** Starting CL353
*** Starting CL354
*** Starting CL355
*** Starting CL356
*** Starting CL357
*** Starting CL358
*** Starting CL359
*** Starting CL360
*** Starting CL361
*** Starting CL362
*** Starting CL363
*** Starting CL364
*** Starting CL365
*** Starting CL366
*** Starting CL367
*** Starting CL368
*** Starting CL369
*** Starting CL370
*** Starting CL371
*** Starting CL372
*** Starting CL373
*** Starting CL374
*** Starting CL375
*** Starting CL376
*** Starting CL377
*** Starting CL378
*** Starting CL379
*** Starting CL380
*** Starting CL381
*** Starting CL382
*** Starting CL383
*** Starting CL384
*** Starting CL385
*** Starting CL386
*** Starting CL387
*** Starting CL388
*** Starting CL389
*** Starting CL390
*** Starting CL391
*** Starting CL392
*** Starting CL393
*** Starting CL394
*** Starting CL395
*** Starting CL396
*** Starting CL397
*** Starting CL398
*** Starting CL399
*** Starting CL400
*** Starting CL401
*** Starting CL402
*** Starting CL403
*** Starting CL404
*** Starting CL405
*** Starting CL406
*** Starting CL407
*** Starting CL408
*** Starting CL409
*** Starting CL410
*** Starting CL411
*** Starting CL412
*** Starting CL413
*** Starting CL414
*** Starting CL415
*** Starting CL416
*** Starting CL417
*** Starting CL418
*** Starting CL419
*** Starting CL420
*** Starting CL421
*** Starting CL422
*** Starting CL423
*** Starting CL424
*** Starting CL425
*** Starting CL426
*** Starting CL427
*** Starting CL428
*** Starting CL429
*** Starting CL430
*** Starting CL431
*** Starting CL432
*** Starting CL433
*** Starting CL434
*** Starting CL435
*** Starting CL436
*** Starting CL437
*** Starting CL438
*** Starting CL439
*** Starting CL440
*** Starting CL441
*** Starting CL442
*** Starting CL443
*** Starting CL444
*** Starting CL445
*** Starting CL446
*** Starting CL447
*** Starting CL448
*** Starting CL449
*** Starting CL450
*** Starting CL451
*** Starting CL452
*** Starting CL453
*** Starting CL454
*** Starting CL455
*** Starting CL456
*** Starting CL457
*** Starting CL458
*** Starting CL459
*** Starting CL460
*** Starting CL461
*** Starting CL462
*** Starting CL463
*** Starting CL464
*** Starting CL465
*** Starting CL466
*** Starting CL467
*** Starting CL468
*** Starting CL469
*** Starting CL470
*** Starting CL471
*** Starting CL472
*** Starting CL473
*** Starting CL474
*** Starting CL475
*** Starting CL476
*** Starting CL477
*** Starting CL478
*** Starting CL479
*** Starting CL480
*** Starting CL481
*** Starting CL482
*** Starting CL483
*** Starting CL484
*** Starting CL485
*** Starting CL486
*** Starting CL487
*** Starting CL488
*** Starting CL489
*** Starting CL490
*** Starting CL491
*** Starting CL492
*** Starting CL493
*** Starting CL494
*** Starting CL495
*** Starting CL496
*** Starting CL497
*** Starting CL498
*** Starting CL499
*** Starting CL500
*** Starting CL501
*** Starting CL502
*** Starting CL503
*** Starting CL504
*** Starting CL505
*** Starting CL506
*** Starting CL507
*** Starting CL508
*** Starting CL509
*** Starting CL510
*** Starting CL511
*** Starting CL512
*** Starting CL513
*** Starting CL514
*** Starting CL515
*** Starting CL516
*** Starting CL517
*** Starting CL518
*** Starting CL519
*** Starting CL520
*** Starting CL521
*** Starting CL522
*** Starting CL523
*** Starting CL524
*** Starting CL525
*** Starting CL526
*** Starting CL527
*** Starting CL528
*** Starting CL529
*** Starting CL530
*** Starting CL531
*** Starting CL532
*** Starting CL533
*** Starting CL534
*** Starting CL535
*** Starting CL536
*** Starting CL537
*** Starting CL538
*** Starting CL539
*** Starting CL540
*** Starting CL541
*** Starting CL542
*** Starting CL543
*** Starting CL544
*** Starting CL545
*** Starting CL546
*** Starting CL547
*** Starting CL548
*** Starting CL549
*** Starting CL550
*** Starting CL551
*** Starting CL552
*** Starting CL553
*** Starting CL554
*** Starting CL555
*** Starting CL556
*** Starting CL557
*** Starting CL558
*** Starting CL559
*** Starting CL560
*** Starting CL561
*** Starting CL562
*** Starting CL563
*** Starting CL564
*** Starting CL565
*** Starting CL566
*** Starting CL567
*** Starting CL568
*** Starting CL569
*** Starting CL570
*** Starting CL571
*** Starting CL572
*** Starting CL573
*** Starting CL574
*** Starting CL575
*** Starting CL576
*** Starting CL577
*** Starting CL578
*** Starting CL579
*** Starting CL580
*** Starting CL581
*** Starting CL582
*** Starting CL583
*** Starting CL584
*** Starting CL585
*** Starting CL586
*** Starting CL587
*** Starting CL588
*** Starting CL589
*** Starting CL590
*** Starting CL591
*** Starting CL592
*** Starting CL593
*** Starting CL594
*** Starting CL595
*** Starting CL596
*** Starting CL597
*** Starting CL598
*** Starting CL599
*** Starting CL600
*** Starting CL601
*** Starting CL602
*** Starting CL603
*** Starting CL604
*** Starting CL605
*** Starting CL606
*** Starting CL607
*** Starting CL608
*** Starting CL609
*** Starting CL610
*** Starting CL611
*** Starting CL612
*** Starting CL613
*** Starting CL614
*** Starting CL615
*** Starting CL616
*** Starting CL617
*** Starting CL618
*** Starting CL619
*** Starting CL620
*** Starting CL621
*** Starting CL622
*** Starting CL623
*** Starting CL624
*** Starting CL625
*** Starting CL626
*** Starting CL627
*** Starting CL628
*** Starting CL629
*** Starting CL630
*** Starting CL631
*** Starting CL632
*** Starting CL633
*** Starting CL634
*** Starting CL635
*** Starting CL636
*** Starting CL637
*** Starting CL638
*** Starting CL639
*** Starting CL640
*** Starting CL641
*** Starting CL642
*** Starting CL643
*** Starting CL644
*** Starting CL645
*** Starting CL646
*** Starting CL647
*** Starting CL648
*** Starting CL649
*** Starting CL650
*** Starting CL651
*** Starting CL652
*** Starting CL653
*** Starting CL654
*** Starting CL655
*** Starting CL656
*** Starting CL657
*** Starting CL658
*** Starting CL659
*** Starting CL660
*** Starting CL661
*** Starting CL662
*** Starting CL663
*** Starting CL664
*** Starting CL665
*** Starting CL666
*** Starting CL667
*** Starting CL668
*** Starting CL669
*** Starting CL670
*** Starting CL671
*** Starting CL672
*** Starting CL673
*** Starting CL674
*** Starting CL675
*** Starting CL676
*** Starting CL677
*** Starting CL678
*** Starting CL679
*** Starting CL680
*** Starting CL681
*** Starting CL682
*** Starting CL683
*** Starting CL684
*** Starting CL685
*** Starting CL686
*** Starting CL687
*** Starting CL688
*** Starting CL689
*** Starting CL690
*** Starting CL691
*** Starting CL692
*** Starting CL693
*** Starting CL694
*** Starting CL695
*** Starting CL696
*** Starting CL697
*** Starting CL698
*** Starting CL699
*** Starting CL700
*** Starting CL701
*** Starting CL702
*** Starting CL703
*** Starting CL704
*** Starting CL705
*** Starting CL706
*** Starting CL707
*** Starting CL708
*** Starting CL709
*** Starting CL710
*** Starting CL711
*** Starting CL712
*** Starting CL713
*** Starting CL714
*** Starting CL715
*** Starting CL716
*** Starting CL717
*** Starting CL718
*** Starting CL719
*** Starting CL720
*** Starting CL721
*** Starting CL722
*** Starting CL723
*** Starting CL724
*** Starting CL725
*** Starting CL726
*** Starting CL727
*** Starting CL728
*** Starting CL729
*** Starting CL730
*** Starting CL731
*** Starting CL732
*** Starting CL733
*** Starting CL734
*** Starting CL735
*** Starting CL736
*** Starting CL737
*** Starting CL738
*** Starting CL739
*** Starting CL740
*** Starting CL741
*** Starting CL742
*** Starting CL743
*** Starting CL744
*** Starting CL745
*** Starting CL746
*** Starting CL747
*** Starting CL748
*** Starting CL749
*** Starting CL750
*** Starting CL751
*** Starting CL752
*** Starting CL753
*** Starting CL754
*** Starting CL755
*** Starting CL756
*** Starting CL757
*** Starting CL758
*** Starting CL759
*** Starting CL760
*** Starting CL761
*** Starting CL762
*** Starting CL763
*** Starting CL764
*** Starting CL765
*** Starting CL766
*** Starting CL767
*** Starting CL768
*** Starting CL769
*** Starting CL770
*** Starting CL771
*** Starting CL772
*** Starting CL773
*** Starting CL774
*** Starting CL775
*** Starting CL776
*** Starting CL777
*** Starting CL778
*** Starting CL779
*** Starting CL780
*** Starting CL781
*** Starting CL782
*** Starting CL783
*** Starting CL784
*** Starting CL785
*** Starting CL786
*** Starting CL787
*** Starting CL788
*** Starting CL789
*** Starting CL790
*** Starting CL791
*** Starting CL792
*** Starting CL793
*** Starting CL794
*** Starting CL795
*** Starting CL796
*** Starting CL797
*** Starting CL798
*** Starting CL799
*** Starting CL800
*** Starting CL801
*** Starting CL802
*** Starting CL803
*** Starting CL804
*** Starting CL805
*** Starting CL806
*** Starting CL807
*** Starting CL808
*** Starting CL809
*** Starting CL810
*** Starting CL811
*** Starting CL812
*** Starting CL813
*** Starting CL814
*** Starting CL815
*** Starting CL816
*** Starting CL817
*** Starting CL818
*** Starting CL819
*** Starting CL820
*** Starting CL821
*** Starting CL822
*** Starting CL823
*** Starting CL824
*** Starting CL825
*** Starting CL826
*** Starting CL827
*** Starting CL828
*** Starting CL829
*** Starting CL830
*** Starting CL831
*** Starting CL832
*** Starting CL833
*** Starting CL834
*** Starting CL835
*** Starting CL836
*** Starting CL837
*** Starting CL838
*** Starting CL839
*** Starting CL840
*** Starting CL841
*** Starting CL842
*** Starting CL843
*** Starting CL844
*** Starting CL845
*** Starting CL846
*** Starting CL847
*** Starting CL848
*** Starting CL849
*** Starting CL850
*** Starting CL851
*** Starting CL852
*** Starting CL853
*** Starting CL854
*** Starting CL855
*** Starting CL856
*** Starting CL857
*** Starting CL858
*** Starting CL859
*** Starting CL860
*** Starting CL861
*** Starting CL862
*** Starting CL863
*** Starting CL864
*** Starting CL865
*** Starting CL866
*** Starting CL867
*** Starting CL868
*** Starting CL869
*** Starting CL870
*** Starting CL871
*** Starting CL872
*** Starting CL873
*** Starting CL874
*** Starting CL875
*** Starting CL876
*** Starting CL877
*** Starting CL878
*** Starting CL879
*** Starting CL880
*** Starting CL881
*** Starting CL882
*** Starting CL883
*** Starting CL884
*** Starting CL885
*** Starting CL886
*** Starting CL887
*** Starting CL888
*** Starting CL889
*** Starting CL890
*** Starting CL891
*** Starting CL892
*** Starting CL893
*** Starting CL894
*** Starting CL895
*** Starting CL896
*** Starting CL897
*** Starting CL898
*** Starting CL899
*** Starting CL900
*** Starting CL901
*** Starting CL902
*** Starting CL903
*** Starting CL904
*** Starting CL905
*** Starting CL906
*** Starting CL907
*** Starting CL908
*** Starting CL909
*** Starting CL910
*** Starting CL911
*** Starting CL912
*** Starting CL913
*** Starting CL914
*** Starting CL915
*** Starting CL916
*** Starting CL917
*** Starting CL918
*** Starting CL919
*** Starting CL920
*** Starting CL921
*** Starting CL922
*** Starting CL923
*** Starting CL924
*** Starting CL925
*** Starting CL926
*** Starting CL927
*** Starting CL928
*** Starting CL929
*** Starting CL930
*** Starting CL931
*** Starting CL932
*** Starting CL933
*** Starting CL934
*** Starting CL935
*** Starting CL936
*** Starting CL937
*** Starting CL938
*** Starting CL939
*** Starting CL940
*** Starting CL941
*** Starting CL942
*** Starting CL943
*** Starting CL944
*** Starting CL945
*** Starting CL946
*** Starting CL947
*** Starting CL948
*** Starting CL949
*** Starting CL950
*** Starting CL951
*** Starting CL952
*** Starting CL953
*** Starting CL954
*** Starting CL955
*** Starting CL956
*** Starting CL957
*** Starting CL958
*** Starting CL959
*** Starting CL960
*** Starting CL961
*** Starting CL962
*** Starting CL963
*** Starting CL964
*** Starting CL965
*** Starting CL966
*** Starting CL967
*** Starting CL968
*** Starting CL969
*** Starting CL970
*** Starting CL971
*** Starting CL972
*** Starting CL973
*** Starting CL974
*** Starting CL975
*** Starting CL976
*** Starting CL977
*** Starting CL978
*** Starting CL979
*** Starting CL980
*** Starting CL981
*** Starting CL982
*** Starting CL983
*** Starting CL984
*** Starting CL985
*** Starting CL986
*** Starting CL987
*** Starting CL988
*** Starting CL989
*** Starting CL990
*** Starting CL991
*** Starting CL992
*** Starting CL993
*** Starting CL994
*** Starting CL995
*** Starting CL996
*** Starting CL997
*** Starting CL998
*** Starting CL999
*** Starting CL1000

```

Fig. 4: Example of Tx(left)-Rx(right) transmission with SDN and GNU Radio enabled: the LEO satellite case.

SDN and GNU radio, together on the quality of satellite communication. The satellite communication parameters are fully monitored by GNU radio blocks, such as : FEC, channel model, frequency ranges, modulations types etc. .as illustrated in Fig. 2.

Measurements are collected using a real time script "shell" and the "iperf" commands under Linux operating systems between two end users. The maintained data throughout the scenarios is stored in a CSV file and treated using MATLAB software. The "Mininet" version used in the experiment is 2.3.0d6, the operating system distribution used is "Ubuntu 16.04", the GNU Radio Companion version is 3.7.9, and Python 3.0 is used as a compiler for the developed APIs.

**Scenario 1: Without SDN**

Traditional terrestrial satellite links are established without SDN, which is built on capabilities envisaged for the improved satellite networks SDN/GNU Radio. This scenario will be used as the benchmark as it shows the performance of the physical layer of a GEO/LEO satellites using GNU radio emulator. All used components to ensure connectivity are illustrated in Fig. 2. Fig. 3 illustrates the successful build of a GEO/LEO satellites transceiver using Python3, before the SDN deployment. We notice the good and correct emulation of the received signal between two IP nodes (end users) as it is shown by the time domain and the frequency domain plot. This is a cross layer solution to send a satellite traffic between two IP nodes using GNU Radio satellite communication chain.

**Scenario 2: With SDN**

It focuses on performing data transmission in a satellite

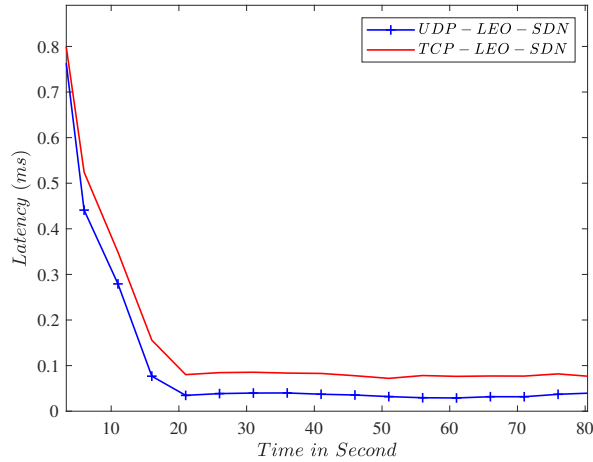


Fig. 6: Latency performance, for LEO sat, with SDN when a streaming load is enabled.

network with SDN as illustrated in Fig. 4 , within the satellite segment on the ground. This scenario demonstrates the integration and the implementation of GEO/LEO satellite communication chain within a SDN network model. A full Southbound API is developed to ensure the integration of GNU radio components illustrated in Fig. 2 to be deployed in an SDN architecture using POX controller.

**Scenario 3: SDN with streaming**

It is build on the capacities envisaged in scenario 2 for the SDN-GNU Radio improved satellite networks. In this use case, a streaming load is added to stress the link and to ensure when a real time networking service is required whether the link is still reliable and stable. This scenario helps to increase the level of Quality of Service-Quality of Experience (QoS-QoE) for the GEO/LEO transmission as shown in Fig. 5. While the left side shows the ongoing transmission, the right side of Fig. 5 shows the perfect reception of the transmitted video streaming.

When it comes to latency as in Fig. 6, TCP protocol shows better performance when SDN is enabled for LEO satellite. In



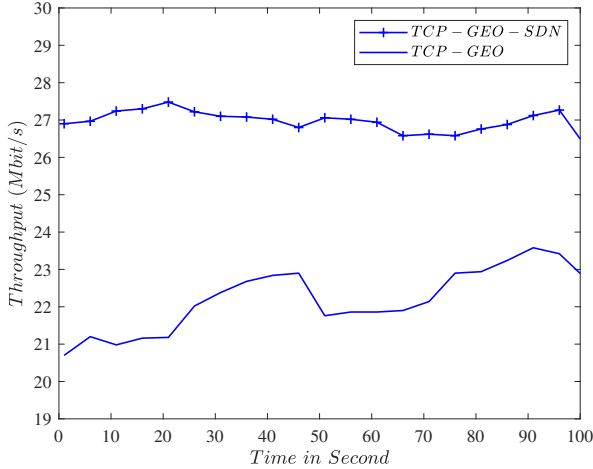


Fig. 7: Throughput performance for GEO sat, with and without SDN under TCP protocol.

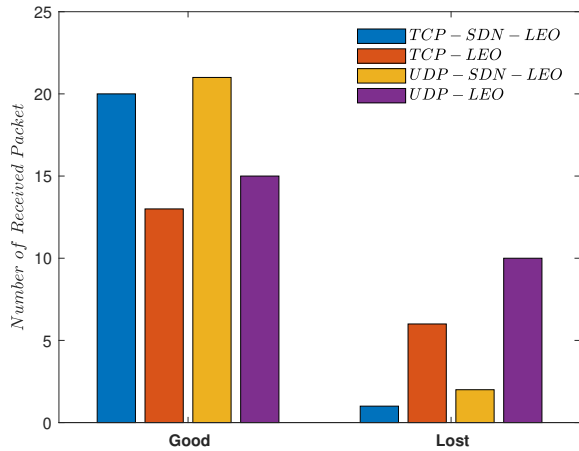


Fig. 8: Packet Loss performance for LEO sat, with and without SDN, under UDP and TCP protocols. Packet Loss statistics, "Good"=no packet lost, "Lost"=packet lost

this use case we added a streaming load as shown in Fig. 5 to see in a closer window the behavior of latency when the link is loaded. It can be noticed, that the time spent in terms of latency when SDN is used becomes ten times higher compared to scenario 1. This could be explained first by the processing time needed by the POX controller to send the required data to the destination. Secondly the southbound API is written in Python3 which is not a Just-In-Time (JIT) compiler.

In Fig. 7, we considered the throughput performance for GEO satellite. An improvement of approximately 6 Mbits/s is recorded, when SDN is applied, compared to scenario 1, when GEO satellite is deployed under TCP protocol only. This could be returned to Openflow protocol for the orchestration of the allocated resources to end users. OpenFlow has an entire control of the network including resource allocation. This can help improving the balancing network environment.

In Fig. 8, we considered the packet transmission perfor-

mance for GEO satellite. The number of good received packets and lost packets is recorded. Statistics shows that the number of good received packets is recorded for TCP and UDP when SDN is enabled (scenario 2). We notice that reliability and connectivity are better ensured under SDN and the controller proves again that it can maintain good connectivity.

#### IV. CONCLUSION

This paper focuses the light on the future of satellite telecommunications. In particular, the combination of SDN and GNU Radio technologies in the field of GEO/LEO satellites is considered as a starting point for making satellite communications possible when integrated into a planned SDN network architecture. Three use cases are described in this paper showing how SDN and GNU Radio technologies can help in the integration and the implementation of space to ground satellite communications. The main scenarios have been identified and analyzed, with an emphasis on improving the infrastructure of the satellite network (scenario 1) and on improving satellite offers for shared terrestrial systems such as communication services by SDN satellite (scenario 2 and 3). Multiple opportunities and benefits have been gained for each scenario, as well as new opportunities through the move from a conventional network to a SDN.

Based on the results of the presented simulations, we found that the architecture was able to obtain better quality of service thanks to an efficient control of satellite traffic under SDN and GNU Radio. However, the paper remains open to possible extensions such as high availability and the use of the P4 language which allows programming of the flow processed by the routing of packets on the transmission equipment without taking into account the format of a packet.

#### ACKNOWLEDGMENT

Authors address a special thanks to Zouhour Raddaoui, Laouini Nour Elhouda from the Higher Institute of Computer Science of Mahdia, Tunisia, and Danial Saleemi for their assistance in Python3 scripting and in the implementation process.

#### REFERENCES

- [1] Y. Gao, H. Ao, Z. Feng, W. Zhou, S. Hu, and X. Li, "Modeling and Practise of Satellite Communication Systems Using Physical Layer Security: A Survey," in *IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2017, Guangzhou, China, July 21-24, 2017, Volume 1*, 2017, pp. 829–832.
- [2] M. R. Haque, S. C. Tan, Z. Yusoff, K. Nisar, C. K. Lee, B. S. Chowdhry, S. Ali, S. K. Memona, and R. Kaspin, "SDN Architecture for UAV's and EVs using Satellite: A Hypothetical Model and New Challenges for Future," in *18th IEEE Annual Consumer Communications & Networking Conference, CCNC 2021, Las Vegas, NV, USA, January 9-12, 2021*. IEEE, 2021, pp. 1–6.
- [3] S. Singh and R. K. Jha, "A Survey on Software Defined Networking: Architecture for Next Generation Network," *CoRR*, vol. abs/2001.10165, 2020.
- [4] E. Kaljic, A. Maric, P. Njemcevic, and M. Hadzialic, "A Survey on Data Plane Flexibility and Programmability in Software-Defined Networking," *IEEE Access*, vol. 7, pp. 47 804–47 840, 2019.
- [5] L. Yan, X. Ding, and G. Zhang, "Dynamic Channel Allocation Aided Random Access for SDN-Enabled LEO satellite iot," *J. Commun. Inf. Networks*, vol. 6, no. 2, pp. 134–141, 2021.
- [6] "OpenFlow switch specification version 1.5.0," available at," Ph.D. dissertation, <http://www.opennetworking.org>, December, 2015.
- [7] X. Hu, Y. Zhang, X. Liao, Z. Liu, W. Wang, and F. M. Ghannouchi, "Dynamic Beam Hopping Method Based on Multi-Objective Deep Reinforcement Learning for Next Generation Satellite Broadband Systems," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 630–646, 2020.

- [8] M. M. Aurizzi, T. Rossi, E. Raso, L. Funari, and E. Cianca, "An SDN-based traffic handover control procedure and SGD management logic for EHF satellite networks," *Comput. Networks*, vol. 196, p. 108260, 2021.
- [9] N. Torkzaban and J. S. Baras, "Controller Placement in SDN-enabled 5G Satellite-Terrestrial Networks," *CoRR*, vol. abs/2108.09176, 2021.
- [10] S. Shi, G. Li, Z. Li, B. Gao, and Z. Luo, "Dynamic Power Allocation Based on Rain Attenuation Prediction for High Throughput Broadband Satellite Systems," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 100-A, no. 9, pp. 2038–2043, 2017.
- [11] T. Ahmed, A. Alleg, R. Ferrús, and R. Riggio, "On-Demand Network Slicing using SDN/NFV-enabled Satellite Ground Segment Systems," in *4th IEEE Conference on Network Softwarization and Workshops, NetSoft 2018, Montreal, QC, Canada, June 25-29, 2018*. IEEE, 2018, pp. 242–246.