

# Background

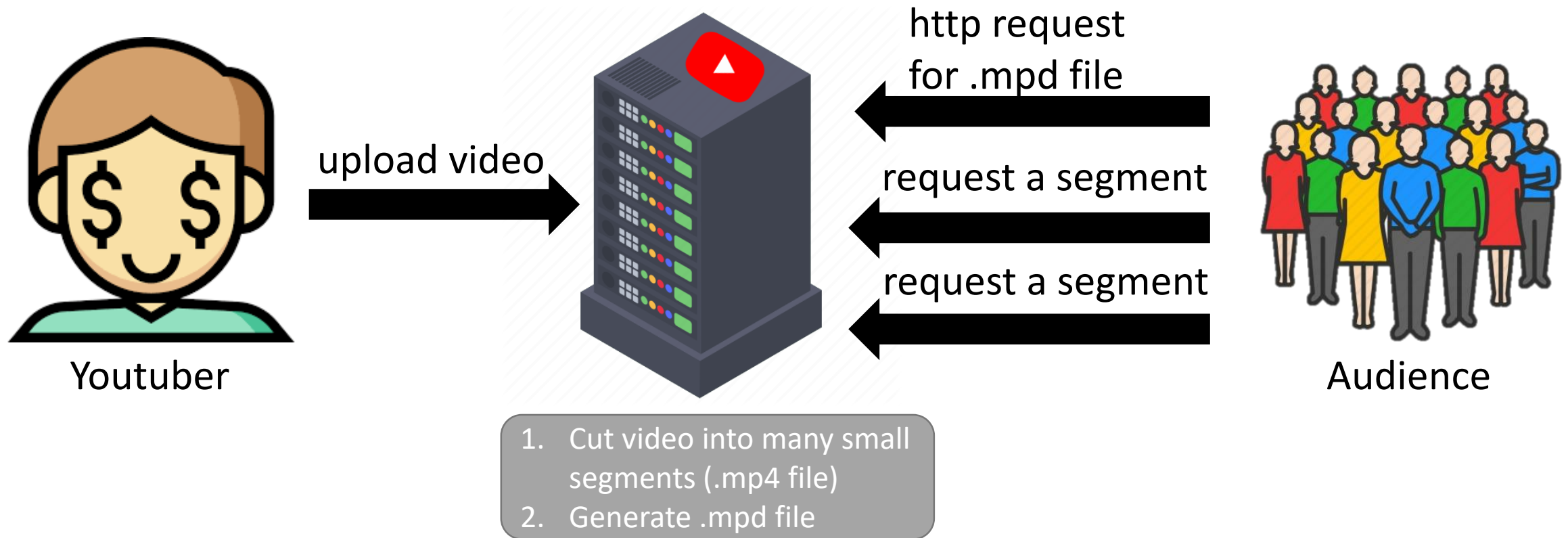
---

Using DASH mechanism for video streaming over LTE architecture

What is DASH (Dynamic Adaptive Streaming over HTTP) ?

- an adaptive bitrate streaming technique.
- enables high quality streaming of media content over HTTP web servers.
- breaking the content into a sequence of small HTTP-based file segments.
- automatically select the segment with the highest bit rate possible that can be downloaded in time for playback without causing stalls or re-buffering events in the playback.

# MPEG-DASH Mechanism



# MPEG-DASH on ns3 (OpenSource)

## Initial Setting:

- users
- Algorithms
- Link Rate
- Buffer Space

## Start:

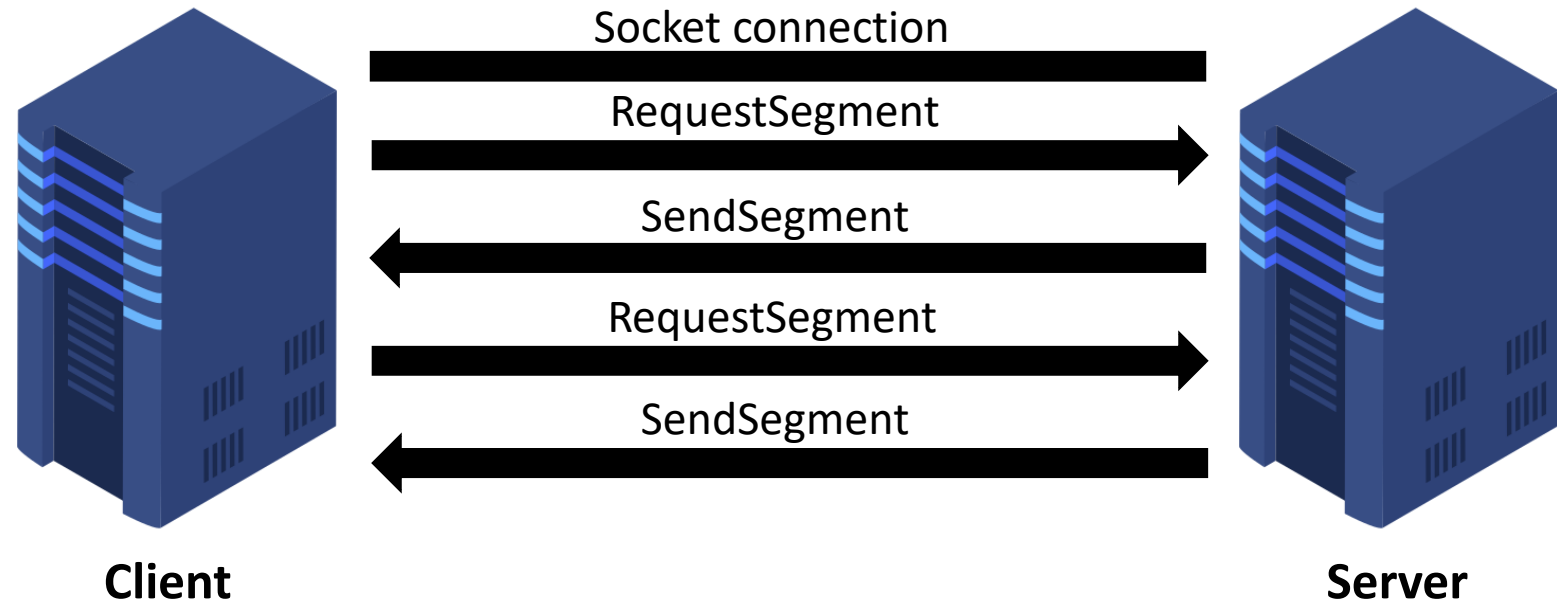
RequestSegment()

## Loop:

- MessageReceived()
- CalcNextSegment()
- RequestSegment()

## Loop:

- HandleRead()
- SendSegment()



# MPEG-DASH - Client

---



Client

Client -> **RequestSegment()**

- Videoid
- bitrate
- segmentId

Client -> **MessageReceived()**

push frame to mpeg-player

if last frame push to frame

- **CalcNextSegment()**
- **RequestSegment()**
- Print mpeg-player info

# MPEG-DASH - Client



Client

## CalcNextSegment()

Use these algorithms to modify bitrate for request segment.

- FDASH
- AAASH
- OSMP
- RAAHS
- SFTM
- SVAA

## FDASH

Input(currRate, nextRate, delay)

Get(m\_target\_dt)

GetBufferEstimate()

GetBufferDifferential()

Output(nextRate)

# MPEG-DASH - Server

---



Server

Server -> **HandleRead()**

- Videoid
- bitrate
- segmentId
- Socket

Server -> **SendSegment()**

Loop (frame):

- Random generate a frame size (according to bitrate)
- Set video play time (50fps)
- Send frame to client

# MPEG-DASH - Log

---

Running:

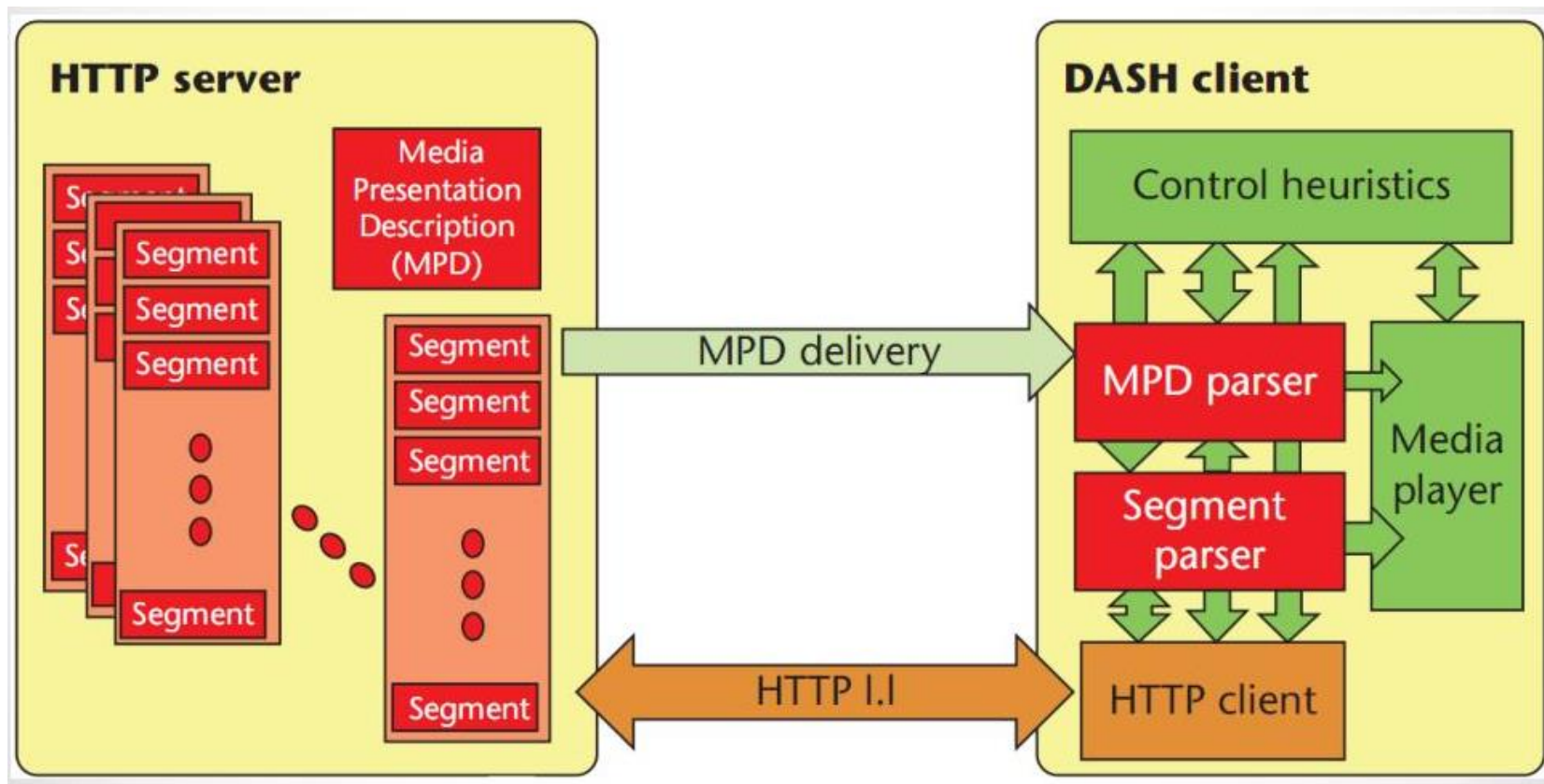
- `newBitRate` : Bitrate of this segment
- `oldBitRate` : Bitrate of previous segment
- `estBitRate` : Estimate next segment bitrate
- `interTime` : Total interrupted time for now
- `T` : How much time is currently available
- `dT` : Currently available time – Previous available time
- `del` : If reach maximum bitrate, need delay time to request next segment

Finally:

- `InterruptionTime` : Total interrupted time
- `Interruptions` : Total number of interruptions
- `avgRate` : Average Bitrate
- `minRate` : Minimum Bitrate
- `AvgDt` : Average currently available time
- `changes` : How much time has bitrate changed?

# MiniProject 3 - Goal

- Simulate DASH mechanism on LTE environment
- Implement a DASH algorithm





# TO-DO

---

1. Download [MPEG-DASH](#) (in /src directory)
  - ./build.py
  - ./waf configure --enable-examples
  - Test :  
./waf --run 'src/dash/examples/dash-example --users=3 --algorithms="ns3::FdashClient" --linkRate=1000Kbps --bufferSpace=100000000'
2. Modify MPEG-DASH examples code “dash-example.cc” to fit LTE environment
3. Design your own algorithms for DASH
  - Reference “./dash/model/algorithms/”
  - Trace the given algorithms and implement a new one.

# Testbed

- Testbed Parameters
  - Streaming time = 30 seconds
  - `./waf --run 'src/dash/examples/dash-lte --users=10 --algorithms="ns3::yourAlgorithm" --linkRate=10Mbps --bufferSapce=100000000'`

- Scoring method for QoE

$$QoE = 5.67 * \frac{\bar{q}}{q_{max}} - 6.72 * \frac{\hat{q}}{q_{max}} + 0.17 - 4.95 * F_i$$

$$F_i = \frac{7}{8} * \max\left(\frac{\ln \phi}{6} + 1, 0\right) + \frac{1}{8} * \left(\frac{\min(\psi, 15)}{15}\right)$$

$\bar{q}$ ,  $\hat{q}$  and  $q_{max}$  are average, standard deviation and max possible of bit rate respectively.  
 $\phi$  and  $\psi$  are freeze frequency and average freeze duration



```

tian@tian-VirtualBox:~/ns-allinone-3.27/ns-3.27$ ./waf --run 'src/dash/examples/dash-lte --users=10 --algorithms="ns3::FdashClient" --linkRate=10Mbps --bufferSpace=10000000'
Waf: Entering directory `/home/tian/ns-allinone-3.27/ns-3.27/build'
Waf: Leaving directory `/home/tian/ns-allinone-3.27/ns-3.27/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (1.995s)
0.677 Node: 0 newBitRate: 131000 oldBitRate: 45000 estBitRate: 282314 interTime: 0 T: 1.303 dT: 0 del: 0
0.678 Node: 1 newBitRate: 131000 oldBitRate: 45000 estBitRate: 271960 interTime: 0 T: 1.302 dT: 0 del: 0
0.68 Node: 2 newBitRate: 131000 oldBitRate: 45000 estBitRate: 289168 interTime: 0 T: 1.3 dT: 0 del: 0
0.68 Node: 3 newBitRate: 131000 oldBitRate: 45000 estBitRate: 268693 interTime: 0 T: 1.3 dT: 0 del: 0
0.682 Node: 4 newBitRate: 131000 oldBitRate: 45000 estBitRate: 279941 interTime: 0 T: 1.298 dT: 0 del: 0
0.683 Node: 7 newBitRate: 131000 oldBitRate: 45000 estBitRate: 293799 interTime: 0 T: 1.297 dT: 0 del: 0
0.687 Node: 8 newBitRate: 131000 oldBitRate: 45000 estBitRate: 275445 interTime: 0 T: 1.293 dT: 0 del: 0
0.692 Node: 5 newBitRate: 131000 oldBitRate: 45000 estBitRate: 272251 interTime: 0 T: 1.288 dT: 0 del: 0
0.695 Node: 6 newBitRate: 131000 oldBitRate: 45000 estBitRate: 266258 interTime: 0 T: 1.285 dT: 0 del: 0
0.699 Node: 9 newBitRate: 89000 oldBitRate: 45000 estBitRate: 259556 interTime: 0 T: 1.281 dT: 0 del: 0

97.32 Node: 4 newBitRate: 178000 oldBitRate: 131000 estBitRate: 211424 interTime: 0 T: 31.964 dT: 0.757 del: 0
97.729 Node: 1 newBitRate: 595000 oldBitRate: 595000 estBitRate: 582684 interTime: 0 T: 37.554 dT: -0.228 del: 0
97.943 Node: 3 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.28472e+06 interTime: 0 T: 41.341 dT: 0.158 del: 0
98.142 Node: 9 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.33208e+06 interTime: 0 T: 41.144 dT: 0.144 del: 0
98.339 Node: 7 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.21138e+06 interTime: 0 T: 40.946 dT: -0.253 del: 0
98.534 Node: 6 newBitRate: 1033000 oldBitRate: 1033000 estBitRate: 1.0387e+06 interTime: 0 T: 44.753 dT: 0.158 del: 0
98.75 Node: 2 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.20158e+06 interTime: 0 T: 44.533 dT: 0.113 del: 0
98.759 Node: 4 newBitRate: 178000 oldBitRate: 178000 estBitRate: 215232 interTime: 0 T: 32.525 dT: 0.561 del: 0
99.121 Node: 5 newBitRate: 522000 oldBitRate: 522000 estBitRate: 535711 interTime: 0 T: 40.165 dT: 0.085 del: 0
99.125 Node: 8 newBitRate: 263000 oldBitRate: 263000 estBitRate: 305600 interTime: 0 T: 34.16 dT: 0.09 del: 0
99.382 Node: 0 newBitRate: 1547000 oldBitRate: 1547000 estBitRate: 1.43161e+06 interTime: 0 T: 39.9 dT: -0.311 del: 0
99.591 Node: 1 newBitRate: 595000 oldBitRate: 595000 estBitRate: 595338 interTime: 0 T: 37.692 dT: 0.138 del: 0
100 Node: 9 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.33926e+06 interTime: 0 T: 41.286 dT: 0.142 del: 0
ns3::FdashClient-Node: 0 InterruptionTime: 0 interruptions: 0 avgRate: 843091 minRate: 131000 AvgDt: 40.8144 changes: 12
ns3::FdashClient-Node: 1 InterruptionTime: 0 interruptions: 0 avgRate: 557646 minRate: 131000 AvgDt: 39.2803 changes: 8
ns3::FdashClient-Node: 2 InterruptionTime: 0 interruptions: 0 avgRate: 534364 minRate: 131000 AvgDt: 44.6958 changes: 8
ns3::FdashClient-Node: 3 InterruptionTime: 0 interruptions: 0 avgRate: 659450 minRate: 131000 AvgDt: 41.6236 changes: 9
ns3::FdashClient-Node: 4 InterruptionTime: 0 interruptions: 0 avgRate: 590291 minRate: 131000 AvgDt: 34.3877 changes: 8
ns3::FdashClient-Node: 5 InterruptionTime: 0 interruptions: 0 avgRate: 348293 minRate: 131000 AvgDt: 38.5204 changes: 14
ns3::FdashClient-Node: 6 InterruptionTime: 0 interruptions: 0 avgRate: 372143 minRate: 131000 AvgDt: 43.6073 changes: 12
ns3::FdashClient-Node: 7 InterruptionTime: 0 interruptions: 0 avgRate: 578376 minRate: 131000 AvgDt: 38.3064 changes: 7
ns3::FdashClient-Node: 8 InterruptionTime: 0 interruptions: 0 avgRate: 430441 minRate: 131000 AvgDt: 33.788 changes: 12
ns3::FdashClient-Node: 9 InterruptionTime: 0 interruptions: 0 avgRate: 653871 minRate: 89000 AvgDt: 42.3353 changes: 10

```



# Project 3

```
tian@tian-VirtualBox:~/ns-allinone-3.26/ns-3.26$ ./waf --run 'src/dash/examples/dash-lte --users=10 --algorithms="ns3::FdashClient" -  
-linkRate=10Mbps --bufferSpace=10000000'
```

```
98.21 Node: 5 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.30848e+06 interTime: 0 T: 43.074 dT: 0.174 del: 0  
98.357 Node: 6 newBitRate: 1033000 oldBitRate: 1033000 estBitRate: 1.13069e+06 interTime: 0 T: 40.928 dT: 0.162 del: 0  
98.367 Node: 8 newBitRate: 396000 oldBitRate: 334000 estBitRate: 394742 interTime: 0 T: 36.919 dT: 0.163 del: 0  
98.901 Node: 7 newBitRate: 1033000 oldBitRate: 1033000 estBitRate: 1.0031e+06 interTime: 0 T: 42.384 dT: -0.307 del: 0  
99.172 Node: 2 newBitRate: 334000 oldBitRate: 334000 estBitRate: 379475 interTime: 0 T: 36.111 dT: 0.196 del: 0  
99.229 Node: 3 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.21999e+06 interTime: 0 T: 38.054 dT: 0.167 del: 0  
99.324 Node: 9 newBitRate: 178000 oldBitRate: 178000 estBitRate: 250747 interTime: 0 T: 31.963 dT: 0.636 del: 0  
99.424 Node: 1 newBitRate: 1245000 oldBitRate: 1245000 estBitRate: 1.29066e+06 interTime: 0 T: 41.858 dT: 0.16 del: 0  
99.99 Node: 4 newBitRate: 1547000 oldBitRate: 1547000 estBitRate: 1.5272e+06 interTime: 0 T: 39.294 dT: 0.133 del: 0  
ns3::FdashClient-Node: 0 InterruptionTime: 0 interruptions: 0 avgRate: 371819 minRate: 131000 AvgDt: 39.0299 changes: 15  
ns3::FdashClient-Node: 1 InterruptionTime: 0 interruptions: 0 avgRate: 601172 minRate: 131000 AvgDt: 41.963 changes: 11  
ns3::FdashClient-Node: 2 InterruptionTime: 0 interruptions: 0 avgRate: 668568 minRate: 131000 AvgDt: 38.3846 changes: 15  
ns3::FdashClient-Node: 3 InterruptionTime: 0 interruptions: 0 avgRate: 566461 minRate: 131000 AvgDt: 34.2981 changes: 7  
ns3::FdashClient-Node: 4 InterruptionTime: 0 interruptions: 0 avgRate: 942791 minRate: 131000 AvgDt: 39.5194 changes: 10  
ns3::FdashClient-Node: 5 InterruptionTime: 0 interruptions: 0 avgRate: 598234 minRate: 131000 AvgDt: 41.2459 changes: 7  
ns3::FdashClient-Node: 6 InterruptionTime: 0 interruptions: 0 avgRate: 467535 minRate: 131000 AvgDt: 38.7941 changes: 11  
ns3::FdashClient-Node: 7 InterruptionTime: 0 interruptions: 0 avgRate: 350400 minRate: 131000 AvgDt: 40.22 changes: 13  
ns3::FdashClient-Node: 8 InterruptionTime: 0 interruptions: 0 avgRate: 380625 minRate: 131000 AvgDt: 37.934 changes: 17  
ns3::FdashClient-Node: 9 InterruptionTime: 0 interruptions: 0 avgRate: 605989 minRate: 131000 AvgDt: 36.1072 changes: 14
```

# Grading Policy

---

- Finish Project 3 (60%)
  - Fit LTE network architecture (30%)
  - Ranking of QoE (30%)
- Report (40%)
  - How do you design your algorithm? (20%)
  - What's the difference between yours and original algorithm? (10%)
  - What you learn? (10%)

# Submission

---

Please upload your miniproject3 to eLearn.

**Deadline: 2020-12-10(Thu.) 23:59**

- Program
  - Your source file must be named as “<Student\_ID>\_project3.cc” and ““<Student\_ID>\_xxxxx-client.cc”.
- Report
  - The report filename must be “<Student\_ID>\_project3.pdf”.

Note: “Plagiarism Avoidance”

- Discussion is encouraged. However, plagiarism is not allowed. We will use, e.g., “Moss” for similarity comparison and 0 points will be given if plagiarism.