TF6
New Transport Protocols

fidel.liberal@ehu.eus
Contributors:
Eneko Atxutegi @KAU @EHU
Åke Arvidsson @ERC
Anna Brunstrom, KJ Grinnemo @KAU
New protocols/services vs TCP

- **LEDBAT**
- **PRR**
- **HTTP/2**
- **QUIC**
  - Overview -> https://peering.google.com/about/quicfaq.html
- **SPUD**
- **CAIA**
- **TAPS**
- **RMCAT**
- **ICCRG**
  - IETF's IRTF -> Being a main goal to "produce an RFC describing the nature of the emerging congestion control problems that any future congestion control architecture must face".

![Protocol Stack Diagram]
Use case: LTE >=4G mobile scenario

• Today
  – Performance of current Transport protocols/CCA over LTE

• New paradigms
  – Caching in eMBMS
  – MEC
Simulated/Emulated environment

If we don't understand the interaction between LTE and different CCA, how can we be able to manage different flows and "decide" the best options for each of them?

Focus on LTE part: implementation and performance study
NS-3 + DCE + LENA project's outcome

• For the model

• For end-nodes (UEs and server)
  • Direct code execution (DCE) -> emulated/virtualised linux machines (from real kernels)
    – Userspace and kernelspace raw code execution
First decisions to simulate a "realistic" scenario

The point was to get drastic changes in terms of throughput. How?

After some discussions and a lot of troubleshooting, we made our first decisions among all the options offered by LTE/EPC module.
First approach

Tests shared points

• Proportional Fair scheduler
• Fading model - EVA (60 km/h)
• 2DRandom mobility model
• Single cell
• 1 UE. AS a receiver (as background traffic). It will also be a foreground traffic (netflix, gaming and so on) receiver.

Tests design

1. As the first approach the UE will move, being its origin the EnodeB.
2. Launch wget traffic as background one. (long file download)
3. Once the flow is established, we will launch out traces, modelling different behaviours (netflix, google maps, gaming, web page and so on).
4. TCP's different flavours are used in this regard.
5. Capture as many metrics as possible to try understanding mostly the impact on throughput and delay.
Preliminary results - Netflix & Cubic

Available vs used bandwidth

Sum of throughputs. Throughput=Bytes/RTT

High probability of packet corruption, retransmissions, drop events, ...

Let's do the same, but with more TCP flavours !!!
Netflix as foreground traffic (I)
10 CCA over Netflix model

No channel losses at RLC level
Macroscopic approach

• Initial conclusions
  – Huge impact of schedulers/mobility patterns/number of users per cell
  – i.e. PF
    • Real achieved throughput or/and consumed RBs
    • Best Transport behaviour may result on lower priority
  – Typical metrics not that useful
• Proportional Fair scheduler
• Fading model - EVA (60 km/h)
• 2DRandom mobility model
• Single cell
• 4 UE. All of them as receivers (as background traffic). One of them will also be a foreground traffic (netflix, gaming and so on) receiver.
Simulations results - Macroscopic view (I)

• 4 UE moving from EnodeB with 2DRandom mobility pattern & 100 packets queue length in EnodeB (I)

As expected, cubic gets the highest throughput but being affected on RTT. Regarding the RTT, westwood performs the best.
Simulations results - Macroscopic view (II)

- 4 UE moving from EnodeB with 2DRandom mobility pattern && 100 packets queue length in EnodeB (II)
Simulations results - Macroscopic view (III)
Next steps

• Refining mobile broadband traffic mix
• Compare existing transport protocols in the emulated scenario
• Test new channel-aware transport protocols
• Include it in new Scenarios
Current Work

4G / 4G+ optimizations based on real-world deployments

QoE-driven (Partial) Channel-awareness

Optimal eNodeB scheduling
max(QoE) subject to sum(RBs)<100
being QoE=f{SBR} and SBR=f{RB,CQI}

GA, MDP, Whittle, Gittings...

QoE-driven
QoE=f{bitrate,energy}

Field Testing
Experimental emulation

CQI traces
1s to 2ms granularity
Future Work

4G+ / 5G optimizations

- Cloud-RAN Mobile Edge Computing
- Partial channel feedback
- Channel feedback
- Optimization placed close to eNodeB
  - eNodeB with less complexity
  - Server instances within / close to RAN
  - eNodeB provides (partial) channel feedback