D4.4

Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network

Abstract:
This Deliverable focuses on the description and performance evaluation of the techniques that have been developed in Task 4.3 in order to enable the efficient delivery of multimedia content over the hybrid satellite-terrestrial backhaul network considered in SANSA. The core of this work focuses on the utilisation of the satellite multicasting communication paradigm, as a means to allow the distribution of popular multimedia content towards the terrestrial content delivery network (CDN) in an effective manner. More specifically, in this Deliverable is described a framework that facilitates the identification of satellite multicast transmission opportunities, the
allocation of the corresponding communication resources (carriers / bandwidth), and the scheduling of the satellite multicast transmissions. In addition, it is proposed a multi-group multicast precoding scheme for a multi-beam full frequency re-use setup with a number of distributed partially cooperating (i.e. exchanging channel state information (CSI) but not user data) satellite gateways (GW). Building on this work, we also present a hybrid satellite-terrestrial proactive caching scheme that enables us to further improve the efficiency of content delivery (in terms of data transport delay, throughput, backhaul / core network bandwidth usage, and content servers load). Furthermore, we propose several cooperative and stand-alone reactive caching techniques that complement the aforementioned proactive caching strategy. Some of these reactive caching variants have also the ability to enable opportunistic joint transmission (JT) through the exploitation of the data stored at the different caching nodes. The numerical simulation results indicate substantial performance gains associated with the abovementioned multicast precoding and proactive / reactive caching methods.
Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Editor</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>24/02/2016</td>
<td>AIT</td>
<td>Document creation. Table of Contents (ToC) and assignment of partners.</td>
</tr>
<tr>
<td>0.2</td>
<td>22/07/2016</td>
<td>AIT</td>
<td>Minor revision of the document’s structure (Sections / Subsections) in line with the on-going work.</td>
</tr>
<tr>
<td>0.3</td>
<td>23/11/2016</td>
<td>AIT</td>
<td>Minor modifications of the ToC (new Sections / Subsections added) in line with the on-going work.</td>
</tr>
<tr>
<td>0.4</td>
<td>02/06/2017</td>
<td>AIT</td>
<td>Finalisation of the ToC.</td>
</tr>
<tr>
<td>0.5</td>
<td>07/06/2017</td>
<td>UL</td>
<td>Contribution from UL.</td>
</tr>
<tr>
<td>0.6</td>
<td>14/06/2017</td>
<td>AIT</td>
<td>Contribution from AIT and AVA.</td>
</tr>
<tr>
<td>0.7</td>
<td>14/07/2017</td>
<td>AVA</td>
<td>Contribution from AVA.</td>
</tr>
<tr>
<td>0.8</td>
<td>17/07/2017</td>
<td>AIT</td>
<td>Contributions and document editing by AIT. Quality Assurance (QA)-ready version.</td>
</tr>
<tr>
<td>0.9</td>
<td>24/07/2017</td>
<td>AIT</td>
<td>Minor corrections / additions based on QA output.</td>
</tr>
<tr>
<td>1.0</td>
<td>28/07/2017</td>
<td>AIT</td>
<td>Final document ready for submission.</td>
</tr>
</tbody>
</table>

Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organisation</th>
<th>Contributions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Konstantinos Ntougias</td>
<td>AIT</td>
<td>Executive Summary; Abstract; Sections 1, 5.1, 5.3, 5.4, 5.5, 6; QA; Documents Editing.</td>
</tr>
<tr>
<td>Dimitrios Ntaikos</td>
<td>AIT</td>
<td>Executive Summary; Abstract; Sections 1, 5.1, 5.3, 5.4, 5.5, 6; QA; Documents Editing.</td>
</tr>
<tr>
<td>George Ziaragkas</td>
<td>AVA</td>
<td>Sections 2, 3; QA.</td>
</tr>
<tr>
<td>Diana Mathew</td>
<td>AVA</td>
<td>Sections 2, 3; QA.</td>
</tr>
<tr>
<td>Eva Lagunas</td>
<td>ULUX</td>
<td>Sections 4, 5.2; QA.</td>
</tr>
<tr>
<td>Ashkan Kalantari</td>
<td>ULUX</td>
<td>Sections 4, 5.2; QA.</td>
</tr>
<tr>
<td>Symeon Chatzinotas</td>
<td>ULUX</td>
<td>Sections 4, 5.2; QA.</td>
</tr>
</tbody>
</table>

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
## Table of Contents

List of Figures............................................................................................................................ 7  
List of Tables.............................................................................................................................. 11  
List of Acronyms and Abbreviations .......................................................................................... 12  
Executive Summary.................................................................................................................... 14  
1 Introduction............................................................................................................................ 18  
  1.1 Background and Motivation ............................................................................................... 18  
  1.2 Contributions of the Deliverable ....................................................................................... 21  
  1.3 Structure of the Deliverable ............................................................................................... 23  
2 Satellite Multicasting: Benefits, Protocols, and Examples.................................................. 23  
  2.1 Transmission Paradigms .................................................................................................... 24  
  2.2 Challenges of Multicast Technologies .............................................................................. 25  
    2.2.1 Multicast in Wired Networks ....................................................................................... 25  
    2.2.2 Multicast in Wireless Communications Networks ....................................................... 25  
    2.2.3 Multicast over Satellite ............................................................................................... 25  
  2.3 Satellite Multicast ............................................................................................................... 26  
    2.3.1 Benefits ....................................................................................................................... 26  
    2.3.2 NORM Protocol ............................................................................................................. 26  
    2.3.3 Examples of Multicast via Satellite ............................................................................. 28  
3 Cooperation Framework that Facilitates Satellite Multicasting ........................................ 31  
  3.1 Motivation .......................................................................................................................... 31  
  3.2 Satellite Capacity Utilisation Statistics ............................................................................. 31  
  3.3 Cooperation Framework Enabling Efficient Resources Allocation and Transmissions  
    Scheduling for Satellite Multicast .......................................................................................... 33  
4 Multicast Multi-group Precoding for the SANSA Satellite Segment with Distributed Gateways 35  
  4.1 Motivation .......................................................................................................................... 35  
  4.2 System model ...................................................................................................................... 36
4.2.1 Multi-Gateway System Model ................................................................. 38
4.2.2 Multi-Beam Satellite Channel ................................................................. 38
4.3 Proposed Multicast Multi-Group Precoding with Distributed Gateways .......... 39
4.4 Numerical Results .................................................................................... 42
4.5 Conclusions ............................................................................................. 46
5 Caching for Hybrid Satellite-Terrestrial Backhaul Networks ............................ 47
5.1 Introduction ............................................................................................. 47
  5.1.1 Mobile Data Traffic Growth ................................................................. 47
  5.1.2 Caching: Concept and Benefits .......................................................... 48
  5.1.3 Caching Efficiency .............................................................................. 48
  5.1.4 Reactive vs. Proactive Caching .......................................................... 49
  5.1.5 Caching Schemes .............................................................................. 49
  5.1.6 Traffic Model ..................................................................................... 49
  5.1.7 Effect of Object Size on Caching Efficiency ....................................... 51
5.2 Hybrid Satellite-Terrestrial Off-Line Caching ................................................ 52
  5.2.1Motivation .......................................................................................... 52
  5.2.2 Hybrid Multi-beam Satellite-Terrestrial SANSA Backhaul Network and the Mono-
      beam Satellite-Terrestrial Benchmark Scheme .................................... 53
  5.2.3 Content Popularity model and local popularity profile generation ......... 57
  5.2.4 Numerical results .............................................................................. 58
  5.2.5 Conclusions ..................................................................................... 64
5.3 Novel Hybrid Reactive Caching Strategy for Hybrid Satellite-Terrestrial Backhaul Networks 65
  5.3.1 Introduction and Motivation ................................................................. 65
  5.3.2 Score-Gated LRU .............................................................................. 66
  5.3.3 Hybrid SG-LRU: Adaptation of SG-LRU for a Network of Caches ....... 74
  5.3.4 Numerical Simulation Results ............................................................. 75
  5.3.5 Conclusions ..................................................................................... 85
5.4 (SG-)LRU-based Cooperative Reactive Caching Schemes ......................................................... 87
   5.4.1 Introduction and Motivation .................................................................................. 87
   5.4.2 A Family of Cooperative Reactive Caching Schemes ......................................... 88
   5.4.3 Numerical Simulation Results .............................................................................. 91
   5.4.4 Conclusions ......................................................................................................... 98

5.5 Cache-Enabled Opportunistic Joint Transmission ......................................................... 102
   5.5.1 Introduction and Motivation ................................................................................. 102
   5.5.2 Numerical results ................................................................................................. 103
   5.5.3 Conclusions ......................................................................................................... 104

6 Summary and Conclusions ............................................................................................... 104

References ............................................................................................................................. 107
List of Figures

Figure 2-1. Broadcast and multicast transmission techniques .................................................. 24
Figure 2-2. Satellite multicast implementation using the NORM protocol in Avanti’s network ..... 27
Figure 2-3. Distribution of UHD films to remote cinemas in the UK ........................................ 29
Figure 2-4. Multicast implementation in the iKnowledge project ........................................... 30
Figure 3-1. Average throughput distribution over a week ....................................................... 32
Figure 3-2. Average throughput distribution over a day .......................................................... 32
Figure 3-3. Distribution of peak hours ..................................................................................... 33
Figure 3-4. Schematic illustrating the proposed multicasting process .................................... 34
Figure 4-1. Beam pattern covering Europe, provided by [28], with the 49 beams considered herein highlighted. Different colours denote separate clusters of beams served by individual GWs, over the same physical layer resources ................................................................. 38
Figure 4-2. Schematic view of the multi-group multicast FL with multiple GWs ....................... 40
Figure 4-3. Minimum SINR over the coverage area versus on-board available transmit power, for 2 backhaul nodes per frame .......................................................................................... 42
Figure 4-4. CDF of backhaul link SINR over the coverage area, for 2 backhaul nodes per frame and per beam available on-board power of 50 Watts ............................................................................. 43
Figure 4-5. CDF of satellite backhaul SINR over the coverage area, for 2 backhaul nodes per frame and per beam available on-board power of 200 Watts ................................................................. 44
Figure 4-6. Power consumption versus per beam available on-board power ............................... 45
Figure 4-7. Energy efficiency versus per beam available on-board power .................................. 46
Figure 5-1. Hybrid satellite-terrestrial backhaul network using a mono-beam satellite .......... 55
Figure 5-2. Hybrid satellite-terrestrial backhaul network using a multi-beam satellite for more accurate content placement ........................................................................................................ 56
Figure 5-3. Schematic of the proposed hybrid satellite-terrestrial backhaul network and the benchmark caching algorithms ........................................................................................................... 57
Figure 5-4. Comparison among the cache hit rate of satellite-only, hybrid satellite-terrestrial, and terrestrial-only backhaul benchmark architectures versus the cache memory size for $\alpha = 0.7$ .... 60
Figure 5-5. Required time for content placement with respect to the cache memory size for mono-beam hybrid satellite-terrestrial, satellite-only, and terrestrial-only benchmark architectures. .... 60
Figure 5-6. Comparison of the cache hit rate between the mono-beam hybrid satellite-terrestrial architectures and the benchmark schemes for different values of $\alpha$ ................................................................. 61
Figure 5-7. Comparison among the cache hit rate of multi-beam and benchmark mono-beam architectures versus the cache memory size for $\alpha = 0.7$ ................................................................................. 62
Figure 5-8. Comparison of the cache hit rate between the multi-beam and the benchmark mono-beam architectures for different values of $\alpha$ ............................................................................... 63
Figure 5-9. Required time for content placement with respect to the cache memory size for the multi-beam hybrid satellite-terrestrial (FFR and conventional), satellite-only, and terrestrial-only architectures.

Figure 5-10. High-level overview of caching operations.

Figure 5-11. Operation of an LRU cache with size $M = 4$. In (1), where a cache miss takes place and the cache storage is not exhausted, the requested object (green) is inserted in the cache and placed on the top of the LRU stack. In (2), where a cache miss takes place and the cache storage is full, the requested object (purple) replaces the LRU cached object (orange). In (3), where a cache hit takes place, the requested object (blue) is placed on top of the cache, since it is the most recently requested (i.e. “most valuable”, according to the LRU principle) object.

Figure 5-12. Comparison of LRU, SW-LFU, and GF-LFU caching schemes. We see that these caching strategies rank differently the objects, despite the fact that the user requests sequence is the same. More specifically, in LRU the object with ID = 7 has the highest rank, since it is the most recently requested object, and it is followed by the objects with ID = 5, ID = 6, and ID = 3 (the LRU object). In SW-LFU, on the other hand, the object with ID = 5 has a score equal to 3 (which could be denoted as $S(5) = 3$), since it has been requested three times in the sliding window of $W = 10$ past requests. Similarly, $S(6) = S(2) = 2$ but the object with ID = 6 has a higher rank than the object with ID = 2, since it has been requested more recently (LRU tie-breaker). Finally, $S(7) = 1$. In GF-LFU, we note that $S(5) = 0.9^3 + 0.9^4 + 0.9^7 = 1.97$. The score of the other objects is calculated in similar manner (e.g. $S(7) = 0.9^0 = 1$). (Adapted from [12].)

Figure 5-13. Operation of an SG-LRU cache with size $M = 4$ that utilises the SW-LFU score function with a sliding window of size $W = 8$. At the left, we note that the requested object (purple) has the same score with the “least valuable” cached object (orange), that is, $S(\text{purple}) = S(\text{orange}) = 1$. Therefore, since it has been more recently requested, it replaces that cached object according to the LRU principle. At the center, where the requested object is the brown one and the “least valuable” cached object is the red one, we note that $S(\text{brown}) = 0 < S(\text{red}) = 1$. Therefore, the requested object does not replace this cached object. Instead, the latter object is placed on the top of the LRU stack, as it is shown at the right.

Figure 5-14. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with $\alpha = 0.50$ against the corresponding LRU and LFU average cache hit rates.

Figure 5-15. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with $\alpha = 0.75$ against the corresponding LRU and LFU average cache hit rates.

Figure 5-16. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with $\alpha = 0.99$ against the corresponding LRU and LFU average cache hit rates.

Figure 5-17. Average cache hit rate of SG-LRU for different Zipf shape factor values against the corresponding LRU and LFU average cache hit rates.
Figure 5-18. Average cache hit rate of SG-LRU for different cache sizes against the corresponding LRU and LFU average cache hit rates. ............................................................................. 79
Figure 5-19. Average cache hit rate of SG-LRU for different catalogue sizes against the corresponding LRU and LFU average cache hit rates. ......................................................... 79
Figure 5-20. Average cache hit rate of Hybrid SG-LRU vs. LRU in a network of $U = 5$ caches for different level of requests’ sharing between the caches and different values of the local-global popularity balance weight against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node. .................................................. 81
Figure 5-21. Average cache hit rate of Hybrid SG-LRU vs. LRU in a network of $U = 5$ caches for different values of the Zipf shape factor against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node. ........................................................................ 82
Figure 5-22. Average cache hit rate of Hybrid SG-LRU in a network of $U = 5$ caches vs. the average cache hit rate of SG-LRU applied in a single-cache system for different values of the Zipf shape factor........................................................................................................... 83
Figure 5-23. Average cache hit rate of Hybrid SG-LRU in a network of $U = 5$ caches vs. the average cache hit rate of SG-LRU applied in a single-cache system for $\alpha = 0.99$ and different values of the weight $t$ .............................................................................. 83
Figure 5-24. Average cache hit rate of Hybrid SG-LRU vs. LRU in a network of $U = 5$ caches for different cache sizes against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node. ............................................................................. 84
Figure 5-25. Average cache hit rate of Hybrid SG-LRU vs. LRU for a system with varying number of caches of fixed cache size against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node. ............................................................................. 85
Figure 5-26. Average cache hit rate of Hybrid SG-LRU vs. LRU for a system with varying number of caches of varying cache size, so that the size of the composite cache is fixed, against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node. ............................................................................. 86
Figure 5-27. Local hit vs. global hit in cooperative caching system with $U = 3$ nodes. The requests arrive at Cache #1. All nodes operate as LRU caches with size $M_i = M_0 = 4$. The caching of objects at the target cache when a global hit occurs is allowed. ...................................................... 88
Figure 5-28. Average composite, local, and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. Zipf exponent for a system setup with $M = 100$, $U = 5$, $M_0 = M/U = 20$, $N = 1,000$ and $W = 10,000$. (a) $\alpha = 0.50$. (b) $\alpha = 0.75$. (c) $\alpha = 0.99$. ................................. 93
Figure 5-29. Average composite, local, and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. composite cache size for a system setup with $\alpha = 0.75$, $U = 5$, $N = 1,000$ and $W = 10,000$. (a) $M = 10$, $M_0 = 2$. (b) $M = 100$, $M_0 = 20$. (c) $M = 1,000$, $M_0 = 200$. .................................................................................................................. 95
Figure 5-30. Average composite, local, and global cache hit rate of SG-LRU for various collaborative caching schemes vs. sliding window size for a system setup with $\alpha = 0.75$, $M = 100$, $U = 5$, $M0 = 20$, and $N = 1,000$. (a) $W = 100$. (b) $W = 1,000$. (c) $W = 10,000$.

Figure 5-31. Average composite, local and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. number of caches for a system setup with $\beta = 0.75$, $M = 100$, $N = 1,000$, and $W = 10,000$. (a) $U = 2$, $M0 = 50$. (b) $U = 5$, $M0 = 20$. (c) $U = 10$, $M0 = 10$...

Figure 5-32. Ratio of JT opportunities for Hybrid SG-LRU and different collaborative caching variants.
List of Tables

Table 4-1. Link budget parameters........................................................................................................ 43
Table 5-1. Collaborative Caching Protocols........................................................................................... 90
List of Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4G</td>
<td>4th Generation</td>
</tr>
<tr>
<td>5G</td>
<td>5th Generation</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BF</td>
<td>Beamforming</td>
</tr>
<tr>
<td>BN</td>
<td>Backhaul Node</td>
</tr>
<tr>
<td>BS</td>
<td>Base Station</td>
</tr>
<tr>
<td>CCI</td>
<td>Co-Channel Interference</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Distribution Network</td>
</tr>
<tr>
<td>CN</td>
<td>Core Network</td>
</tr>
<tr>
<td>CO</td>
<td>Central Office</td>
</tr>
<tr>
<td>CoMP</td>
<td>Coordinated Multi-Point</td>
</tr>
<tr>
<td>CSI</td>
<td>Channel State Information</td>
</tr>
<tr>
<td>CSIT</td>
<td>CSI at the Transmitter</td>
</tr>
<tr>
<td>DL</td>
<td>Downlink</td>
</tr>
<tr>
<td>DVB-S</td>
<td>Digital Video Broadcasting Satellite</td>
</tr>
<tr>
<td>eMBB</td>
<td>Extreme MBB</td>
</tr>
<tr>
<td>FBP</td>
<td>Frame-Based Precoding</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FL</td>
<td>Forward Link</td>
</tr>
<tr>
<td>GF</td>
<td>Geometric Fading</td>
</tr>
<tr>
<td>GW</td>
<td>Gateway</td>
</tr>
<tr>
<td>HNM</td>
<td>Hybrid Network Manager</td>
</tr>
<tr>
<td>HPA</td>
<td>High Power Amplifier</td>
</tr>
<tr>
<td>HTS</td>
<td>High Throughput Satellite</td>
</tr>
<tr>
<td>IBI</td>
<td>Inter-Beam Interference</td>
</tr>
<tr>
<td>iBN</td>
<td>Intelligent Backhaul Node</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>independent and identically distributed</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IRM</td>
<td>Independent Reference Model</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter-Symbol Interference</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>JT</td>
<td>Joint Transmission</td>
</tr>
<tr>
<td>LFU</td>
<td>Least Frequently Used</td>
</tr>
</tbody>
</table>

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]  
Date: 28/07/2017
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU</td>
<td>Least Recently Used</td>
</tr>
<tr>
<td>LTE</td>
<td>Long Term Evolution</td>
</tr>
<tr>
<td>MBB</td>
<td>Mobile Broadband</td>
</tr>
<tr>
<td>MBH</td>
<td>Mobile Backhaul</td>
</tr>
<tr>
<td>MBSFN</td>
<td>Multicast-Broadcast Single Frequency Network</td>
</tr>
<tr>
<td>MSP</td>
<td>Multicast Service Provider</td>
</tr>
<tr>
<td>NACK</td>
<td>Negative Acknowledgement (ACK)</td>
</tr>
<tr>
<td>NORM</td>
<td>Negative-acknowledgement Oriented Reliable Multicast</td>
</tr>
<tr>
<td>Over The Top</td>
<td>OTT</td>
</tr>
<tr>
<td>P2P</td>
<td>Peer-to-Peer</td>
</tr>
<tr>
<td>PAC</td>
<td>Per-Antenna power Constraints</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical layer</td>
</tr>
<tr>
<td>pmf</td>
<td>probability mass function</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality-of-Service</td>
</tr>
<tr>
<td>RAN</td>
<td>Radio Access Network</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>SCN</td>
<td>Small Cell Network</td>
</tr>
<tr>
<td>SE</td>
<td>Spectral Efficiency</td>
</tr>
<tr>
<td>SG</td>
<td>Score Gate</td>
</tr>
<tr>
<td>SINR</td>
<td>Signal-to-Interference-plus-Noise-Ratio</td>
</tr>
<tr>
<td>SM</td>
<td>Spatial Multiplexing</td>
</tr>
<tr>
<td>SPC</td>
<td>Sum Power Constraints</td>
</tr>
<tr>
<td>SR</td>
<td>Sum-Rate</td>
</tr>
<tr>
<td>SSP</td>
<td>Satellite Service Provider</td>
</tr>
<tr>
<td>SW</td>
<td>Sliding Window</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TFMCC</td>
<td>TCP-Friendly Multicast Congestion Control</td>
</tr>
<tr>
<td>TV</td>
<td>Television</td>
</tr>
<tr>
<td>UHD</td>
<td>Ultra-High-Definition</td>
</tr>
<tr>
<td>UT</td>
<td>User Terminal</td>
</tr>
<tr>
<td>VoD</td>
<td>Video on Demand</td>
</tr>
<tr>
<td>VSAT</td>
<td>Very Small Aperture Terminal</td>
</tr>
</tbody>
</table>
Executive Summary

The emergence of video streaming services resulted in an enormous growth of the mobile data traffic’s volume over the past few years. This trend is expected to continue in the foreseeable future, as the quality of multimedia content gets higher and challenging use cases, such as the delivery of video streams at locations with high user density, are emerging. The exponential increase of the mobile traffic places a heavy burden on the backhaul.

The use of hybrid satellite-terrestrial backhaul networks, as the one considered in SANSA, can address this issue. In such systems, the satellite capacity is exploited to offload the terrestrial backhaul. The inherent multicast nature of satellite communications enables the transmission of sets of common data to distinct groups of user terminals. This communication paradigm, which is referred to as multi-group multicasting, allows the efficient distribution of popular multimedia content towards the terrestrial content delivery network (CDN). For satellite multi-group multicast transmission to take place, the satellite service provider (SSP) and the multicast service provider (MSP) should cooperate closely.

The state-of-the-art in satellite communications technology refers to a multi-beam architecture, which enables the application of aggressive frequency re-use across the service area. While today it is common a “four colour” frequency re-use pattern, next generation systems are expected to utilise full frequency re-use schemes, in response to the increased capacity requirements. In such setups, joint multi-beam precoding should be applied, in order to mitigate the inter-beam interference (IBI) caused by the transmission of co-channel signals at adjacent spot beams. This precoding scheme should present a number of characteristics and consider various practical limitations:

- It should be multi-group multicast aware.
- It should preserve the framing of the utilised protocol (e.g. DVB-S2x), which has been optimised according to the delay and transmission power limitations of satellite communications.
- Since typically a single feed is used per spot-beam, it should provide optimal performance under the consideration of per-antenna power constraints (PAC) instead of sum-power constraints (SPC).
- In practice, a number of interconnected satellite gateways (GWs) is deployed, in order to alleviate the capacity limitations of the feeder link.
- Typically, partial cooperation between the satellite GWs is employed instead of a full cooperation scheme (i.e. the GWs share their local Channel State Information (CSI) but they
do not share the user data), due to the strict capacity and synchronisation requirements of the latter approach.

The backhaul nodes (BN) can also operate as caches, in order to further offload the terrestrial backhaul and reduce the data transport delays. Assuming the application of proactive caching, satellite multi-group multicasting can be utilised as a means to efficiently update the cache storage during off-peak hours (e.g. overnight).

Often, reactive caching is integrated with proactive caching, in order to further enhance the performance of the system by combining the best elements from these two “worlds” and eliminating their drawbacks. The most commonly applied reactive caching scheme is Least Recently Used (LRU), which ranks the requested pieces of content (also referred to as “objects”) according to their time-of-last-access and holds in the cache storage the most recently requested objects. Thus, it can be efficiently implemented in software as a doubly-linked list which always places the requested objects on the top of a stack of cached objects sorted according to their time-of-last access. LRU is the “de facto standard” reactive caching strategy due to its simple implementation, low and constant cache update effort per request, and ability to react to the content popularity dynamics noticed in practice. However, it suffers from frequent loading of objects into the cache and storage of “one-timers” that degrade the caching efficiency. Moreover, it is easily proven, in view of the Zipf-like requests patterns noticed in practice, that its performance is suboptimal, due to the fact that it does not exploit content popularity information (i.e. object requests frequency) in the caching decisions. Least Frequently Use (LFU) represents the other extreme in the caching policies spectrum. This strategy ranks the requested objects according to their request count. Under the Independent Reference Model (IRM), LFU achieves the optimum cache hit rate. On the other hand, the use of an infinite backlog of object requests frequency results in pollution of the cache by outdated, formerly popular content as well as in extreme demands in computational and storage resources which turn this caching scheme impractical.

Also, cooperative caching has been recently proposed as a method to offload the backhaul in small cell network (SCN) setups. Different levels of content replication among the caching nodes result in different behaviour and caching efficiency.

In addition, several content placement mechanisms that promote the duplication of content at different cache servers in order to enable opportunistic joint transmission (JT), so that the system benefits from the provided spatial multiplexing (SM) gains while at the same time eliminates the corresponding data sharing overhead, have been studied lately in the literature.

Based on the above discussion, this Deliverable describes novel solutions for each one of these items. More specifically:
A cooperation framework that facilitates the application of satellite multi-group multicasting in the considered hybrid satellite-terrestrial backhaul network is proposed. This framework is based on the exchange of messages between the SSP and the MSP that contain information such as satellite capacity utilisation statistics, beam coverage, throughput, free capacity etc. in order to identify satellite multi-group multicast transmission opportunities, allocate the corresponding communication resources (carriers / bandwidth), and schedule the multicast transmissions.

Focusing on the capacity limitations of current feeder links together with the satellite per-antenna power constraints, we investigate a multi-group multicast precoding scheme for distributed GWs, where only CSI is shared among them. As expected, multi-GW configurations result in inter-cluster interference which severely degrades the system performance. Still, considerable gains over a system without any coordination are reported and in low power regimes, the proposed coordination attains an energy efficiency performance close to the ideal feeder link.

A hybrid satellite-terrestrial proactive caching scheme, which utilises both satellite multi-group multicast and terrestrial unicast transmissions in the content placement phase, is proposed. This caching strategy exploits both local and global content popularity information in content placement decisions. The local popularity information for an object is obtained from the caching node of interest whereas the global one is calculated as the average of the corresponding local popularities over either all the other caching nodes or the remaining caches in the same spot beam (cluster), depending on whether a mono-beam or a multi-beam architecture is adopted. The simulation results revealed that this caching technique reduces significantly the content placement delay while it approaches closely the optimum caching efficiency accomplished in a terrestrial-only setup. Moreover, the simulations indicated that the multi-beam architecture outperforms the mono-beam one, due to the higher accuracy accomplished in global popularities calculation.

A Hybrid Score-Gated LRU (SG-LRU) caching scheme that complements the aforementioned proactive caching strategy is proposed. SG-LRU is an extension of conventional LRU that makes use of an LRU cache and a Sliding Window LFU (SW-LFU) score-gate function. The LRU cache accounts for the simple implementation and cache updates. The score-gate function restricts the LFU principle in a SW of past requests. Thus, (a) it provides an aging mechanism to enable adaptation to content popularity shifts; (b) it provides an admission control mechanism to undercut the loading rate of content into the cache in comparison to standard LRU and avoid the caching of “one-timers”; and (c) it collects into the cache the most popular objects (in a recent timeframe). Changing the size of the SW, SG-LRU may
behave similar to LRU or to LFU or it may lie somewhere in between. Hybrid SG-LRU is an adaptation of SG-LRU for multi-cache setups where the caches are divided into disjoint clusters. This variant of SG-LRU makes use of both local and global scores in order to calculate a total score for each object that will affect the caching decisions. The local score of an object is calculated at each cache according to the SW-LFU function. The global score for this object is the average of its local scores over all caches located in the same cluster. A weight factor determines the local score – global score balance in the calculation of the total score of each object. Simulation results showed that Hybrid SG-LRU outperforms significantly the LRU caching strategy, especially for small / moderate cache size, highly skewed content popularity distribution, and large SW, and approaches closely the LFU cache hit rate.

- A family of LRU / SG-LRU-based collaborative caching schemes that differ on whether they allow or prohibit the update of the cache or / and the SW of the target or / and the remote cache upon a global hit is proposed. Simulation results showed that the SG-LRU subset outperforms significantly its LRU counterparts and approaches the caching efficiency of LFU when the size of the SW is appropriately selected.

- Hybrid SG-LRU and several Collaborative SG-LRU variants create JT opportunities, due to either the convergence of the cached contents based on the utilisation of global popularity statistics or the use of cooperation directives that enforce the replication of content across several caching nodes. Numerical simulations have demonstrated this fact.

To summarise, this Deliverable presents a complete solution for the efficient distribution / delivery of popular multimedia content over the SANSA network based on satellite multi-group multicasting and integrated proactive / reactive (non-cooperative or cooperative) caching. The numerical simulation results demonstrate that the use of the proposed methods offers substantial performance gains.
1 Introduction

1.1 Background and Motivation

As described in [1], SANSA envisions a hybrid terrestrial / satellite mobile backhaul (MBH) network, where satellite links act either as an enhancement of the terrestrial microwave links in order for the composite system to accommodate the predicted traffic demands or as a backup solution in case of link failures or congestion occurring at the terrestrial segment. To this end, certain intelligent backhaul nodes (iBN) support both types of backhauling. In addition, the use of smart antennas by all terrestrial BNs allows for the reconfiguration of the terrestrial segment’s topology. A hybrid network manager (HNM) takes advantage of these features to determine the optimal backhaul routes that will be utilised at each given time over the hybrid backhaul network, so that the available resources are utilised optimally while at the same time some predefined performance criteria are met. Moreover, the application of advanced interference management techniques enables the sharing of the spectrum between the terrestrial and the satellite links, so that the efficiency of the available spectral resources’ utilisation is improved.

As it has been already implied, one of the main reasons for integrating satellite communications technology with the terrestrial backhaul is the need to meet the extremely high capacity requirements of contemporary and next-generation cellular systems, which are mainly attributed to data-hungry services involving the distribution of high-quality multimedia content [2].

Satellite multi-group multicasting, where independent sets of common data are transmitted to distinct groups of users [3], is able to facilitate the efficient delivery of popular multimedia content towards the terrestrial distribution network considered in SANSA.

SANSA assumes, in line with the current trend, that the satellite segment makes use of multi-beam technology in the forward link’s (FL) downlink (DL) (i.e. the link from the satellite station to the terminal device), in response to the ever-increasing demands for high throughput. In this transmission paradigm, which has been inspired by the cellular principle, a large number of spot beams is employed to provide coverage at the service area of interest instead of a single broad beam. Hence, the same frequencies can be re-used at sufficiently separated geographical regions illuminated by different spot beams, so that the resulting inter-beam interference (IBI) is negligible and the overall spectral efficiency (SE) is enhanced. Typical implementations involve tens or hundreds of spot beams and rely on a so-called four colour frequency re-use pattern, where the “colours” correspond to four different frequencies or, more commonly, to two distinct frequencies and two orthogonal polarisations [4].
However, the enormous capacity requirements of future cellular systems dictate the need for more aggressive frequency re-use, leading eventually to a full frequency re-use scheme whereby each spot beam across the coverage area exploits the whole available bandwidth and employs both types of polarisation. On the other hand, this setup introduces IBI due to the transmission of co-channel signals at adjacent spot beams, which can reduce considerably the achieved sum-rate (SR) throughput and degrade the quality-of-service (QoS) provided to the beam-edge user terminals (UT) if it is not accounted for. Therefore, the application of joint multi-beam precoding is considered as a means to cope with the IBI under this scenario [5]. This concept presents many similarities with the joint multi-cell precoding variant of coordinated multi-point (CoMP) [6]. Nevertheless, there exist also substantial differences between these approaches, which refer mainly to implementation details that reflect the nature of the underlying systems. The following list summarises the main similarities and differences between the abovementioned concepts:

- Multi-cell precoding requires the sharing of user data or channel state information (CSI) among the cooperating base stations (BS) over the MBH. Typically, a Cloud RAN architecture is adopted to facilitate the exchange of these signals. In this setup, the relevant information is gathered at a central office (CO), where the precoding operation takes place. Similarly, in multi-beam precoding, each user’s signal is precoded at the satellite gateway (GW) and transmitted by the satellite station over all beams.

- The precoding techniques considered in joint multi-cell precoding can be extended to joint multi-beam precoding.

- The increase in the capacity of the user link requires a corresponding increase in the capacity of the feeder link. In principle, the exploitation of higher frequency bands by this wireless link could address this issue. However, often this approach is not feasible in practice. A common alternative is the deployment of multiple GWs, where each one precodes the signals that will be transmitted through a cluster of neighbouring spot beams. This approach is similar to the use of cell clustering and multiple COs in Cloud RAN setups utilising CoMP transmission and gives rise to distributed joint precoding techniques. The performance degradation of such methods, in comparison with centralised joint precoding, can be mitigated by assuming interconnected GWs and enabling the sharing of both user data and CSI among them, thus reducing the system to a setup with a virtual centralised GW. Nevertheless, the excessive cooperation latency of this full cooperation scheme makes partial cooperation techniques, where only the local CSI is exchanged between the GWs, preferable in such decentralised settings [5].

- The acquisition of CSI at the transmitter (CSIT) is accomplished through pilot-assisted channel estimation at the UTs and report of the channel estimates to the respective GWs.
In general, the required CSIT might be outdated, due to the long delays involved in satellite transmission. This fact could degrade the performance of the system. However, in the scenario considered in SANSA where the terminals are fixed nodes, the wireless channel remains relatively constant over long periods of time. Thus, the aforementioned issue has negligible effect on the system’s performance.

- The current trend in multi-beam satellite communications technology is the use of a single feed per spot beam. This fact implies that a single radio frequency (RF) chain drives each such feed. As a consequence, the corresponding optimisation problems regarding precoding design and power allocation (e.g. SR throughput maximisation, max-min fairness etc.) should take into account per-antenna power constraints (PAC) instead of assuming sum-power constraints (SPC).

- Perhaps most importantly, the multicast nature of satellite communications should be considered in precoding design. Moreover, the framing structure of the various satellite communications standards, which has been optimised according to the delay and transmission power limitations of such systems, motivates the utilisation of frame-based precoding (FBP), where the precoding operations do not affect the underlying frame structure [7].

We should note that in order to fully exploit the multicast capability of satellite communications technologies, we should employ mechanisms that allow us to (a) identify multicasting opportunities; (b) allocate optimally the available carriers / bandwidth, based on relevant capacity utilisation forecasts; and (c) schedule the multi-group multicasting transmissions.

Multicasting enables us also to use proactive caching, in order to further off-load the backhaul and reduce the communication delay. In this case, multi-group multicasting is used as a means to efficiently update the local storage of the deployed caching servers (which are typically collocated with the BNs) at regular intervals (e.g. overnight) [8].

Typically, reactive caching is used in conjunction with proactive caching, in order to further enhance the performance of the installed caching system [9]. Least Recently Used (LRU) is the most commonly applied reactive caching strategy, due to its simple implementation, low computational and storage space requirements, and ability to avoid the pollution of the cache with outdated content. However, it has been manifold demonstrated (analytically as well as via numerical simulations based either on synthetic request patterns or on data traces) that it is suboptimal, in terms of the achieved caching efficiency [10]-[14]. This is because its caching decisions are based solely on content recency information, while it has been proven that content popularity (i.e. content requests frequency) should also be exploited in order to achieve the optimum caching performance.
Another paradigm that has been recently emerged as a means to improve the performance of caching systems installed in small cell networks (SCN) is cooperative caching [15]. The method of cooperation affects the placement of content in the cache servers, leading to different levels of content replication throughout the caching nodes, which might boost or degrade the performance of the system, depending on the behaviour of the users.

Also, it is often desirable to perform content placement in such a manner that Joint Transmission (JT) opportunities are created, so that the SE of the radio access segment is improved through spatial multiplexing (SM) while at the same time the cooperation overhead induced by the requirement of JT for data sharing among the cooperating nodes over the MBH is eliminated [15][16].

1.2 Contributions of the Deliverable

The contributions of this deliverable are summarised in the following list:

• Satellite capacity utilisation statistics obtained from Avanti’s multicast-enabled VSAT network are presented. These data show interesting trends in user behaviour.

• A framework for the efficient cooperation between the satellite service provider (SSP) and the multicast service provider (MSP) in a hybrid satellite-terrestrial network is described. This framework facilitates the optimal exploitation of the satellite’s multicasting capabilities. It is based on the utilisation of parameters such as user demands, carrier / bandwidth utilisation forecasts based on history log statistics etc., in order to enable the identification of multicast opportunities and facilitate the resources allocation and multicast transmissions scheduling procedures.

• A multi-group multicast aware method is proposed and evaluated through numerical simulations for a scenario where a full frequency re-use multi-beam architecture is utilised. This study considers PAC instead of SPC and takes into account both the DVB-S2x framing structure as well as the feeder link’s capacity limitations. Regarding the latter aspect, it is assumed that a number of satellite GWs is distributed across the terrestrial segment as well as that spot beam clustering is employed. The GWs share their local CSI but not user data. The simulation results indicated that while the inter-cluster interference affects significantly the fairness of the system, the proposed algorithm provides good availability to the BNs while offloading the feeder link. In addition, it is shown that when the interference from adjacent clusters is low (e.g. in the low power regime), the proposed coordination scheme attains an energy efficiency performance that approaches the one of the single GW upper bound.
• A hybrid satellite-terrestrial proactive caching technique is proposed and evaluated through numerical simulations, assuming both mono-beam and multi-beam architectures. This caching scheme utilises both satellite multi-group multicast and terrestrial unicast transmissions in the content placement phase. In addition, it makes use of both local popularity information, which is obtained at each cache server (BN), as well as global popularity information, which is calculated per beam as the average of the local popularities of the caching nodes located at this beam, in content placement decisions. The simulation results showed that the proposed hybrid satellite-terrestrial architecture is able to considerably reduce the required time for file placement while achieving a caching that closely approaches the terrestrial-only upper bound. In addition, they indicated that the multi-beam setup outperforms the mono-beam one.

• A novel reactive caching scheme, called Hybrid Score-Gated LRU (H-SG-LRU), that complements the aforementioned proactive caching strategy is proposed and evaluated through numerical simulations. This caching scheme is an adaptation of SG-LRU, which in turn is an extension of LRU that takes into account both time-of-last-access and frequency information in caching decisions, for a scenario where multiple caching nodes are deployed. The term “Hybrid” in this case refers to the fact that this caching strategy exploits both local and global popularity information, assuming a cache clustering scheme, in order to decide whether to cache or not the requested content as well as which cached content should be evicted from the cache storage. The impact of local and global popularity in caching decisions is determined via a weight. SG-LRU presents similar implementation complexity, computation burden, and storage space requirements to the “de facto standard” LRU scheme. Also, it reacts to changes in content popularity, thanks to the use of an aging mechanism. On the other hand, it reduces the rate of loading content into the cache, which increases the computational effort, in comparison to LRU and avoids the storage of “one-timers” (i.e. objects that are requested only one) which degrade the caching efficiency, thanks to the utilisation of admission control. In addition, the simulation results showed that SG-LRU outperforms significantly LRU and approaches the optimum, under a model that assumes constant content popularities, caching efficiency, which is achieved with the infeasible in practical applications, due to its extreme computational complexity and storage capacity requirements, Least Frequently Used (LFU) strategy.

• A family of cooperative caching techniques based on SG-LRU is proposed and evaluated through numerical simulations. The simulation results indicated that Cooperative SG-LRU outperforms significantly its LRU counterparts and approaches the optimum LFU caching efficiency.
In all cases, the simulation results provided useful insights regarding the impact of various parameters on the performance of the proposed methods, which in turn led to corresponding design guidelines.

Finally, the ability of Hybrid SG-LRU and Cooperative SG-LRU to create JT opportunities is demonstrated through numerical simulations.

1.3 Structure of the Deliverable

The structure of this Deliverable is as follows: In Section 2 the benefits of satellite multicasting are discussed and relevant protocols and examples of applications are presented. In Section 3 are provided satellite capacity utilisation statistics, obtained by Avanti’s multicast-enabled VSAT network, and it is presented a cooperation framework between the SSP and the MSP that facilitates satellite multicasting. In Section 4 is presented and evaluated through numerical simulations a multi-group multicast aware FBP method that considers PAC and takes into account the feeder’s link limitations, in a scenario assuming a full frequency re-use multi-beam architecture, multiple GWs that exchange only CSI, and beam clustering. In Section 5 is given a brief overview of caching and then are studied several proposed caching strategies for hybrid satellite-terrestrial networks, namely, a hybrid satellite-terrestrial proactive caching scheme; Hybrid SG-LRU; and Cooperative SG-LRU. Different architectures (mono-beam vs. multi-beam); levels of user requests sharing at different caching nodes; and cooperation variants, respectively, as well as various parameter settings (e.g. Zipf exponent, cache size, number of caching nodes etc.) are considered in the numerical simulations that are used to evaluate the performance of the aforementioned caching methods. Also, the ability of Hybrid SG-LRU and Cooperative SG-LRU to create JT opportunities is demonstrated through numerical simulations. Finally, Section 6 provides the summary and conclusions of our work.

2 Satellite Multicasting: Benefits, Protocols, and Examples

This Section describes briefly the various transmissions paradigms, the challenges presented when applying multicast transmission technologies, and the benefits of employing satellite multicast. In addition, it presents some satellite multicasting protocols and provides examples of satellite multicast applications.
2.1 Transmission Paradigms

There exist three transmission paradigms:

- **Unicast**: Unicast refers to one-to-one transmission from a single source to a single destination (e.g. downloading a web page from a server to a user’s browser).

- **Broadcast**: Broadcast refers to one-to-many transmission from a single source to all possible destinations (e.g. from a satellite to all receivers within a satellite spot beam).

- **Multicast**: Multicast refers to one-to-many transmission or many-to-many from a single or multiple sources to a subset of the possible destinations (e.g. IP-TV content distribution, where a single source multicasts content to a group of receivers).

Multicast presents the following advantages over unicast:

- **Reduced network bandwidth usage**: Multicasting a single data packet to \( N \) recipients, and therefore making multiple copies of this packet only when it has to traverse different links in order to reach all intended destinations as it is seen in Figure 2-1, provides a \( 1/N \) savings in bandwidth usage in comparison to the scenario where \( N \) unicast transmissions take place. This is particularly beneficial in satellite transmission, where the communication resources are limited and expensive.

- **Reduced source processing load**: Regardless of whether “best effort” or “reliable” multicast transmission mode is utilised, the multicast source does not need to maintain state information about the communication link to each individual recipient.
2.2 Challenges of Multicast Technologies

Multicast is currently implemented in terrestrial (wired and wireless) and satellite communication networks, as a means to support the provision of streaming services.

2.2.1 Multicast in Wired Networks

The demand for video traffic is on the rise and is expected to grow manifolds in the foreseeable future. In response to the traffic growth, the network operators and Internet Service Providers (ISP) should upgrade or extend their network, in order to avoid the occurrence of discrepancies between the required and the provided capacity at some network segment. This procedure implies usually the need for substantial investments in network infrastructure. For instance, in order to support multicasting as a means to improve the utilisation efficiency of the available resources, network operators / ISPs have to install multicasting-capable hardware.

2.2.2 Multicast in Wireless Communications Networks

The use of multicasting in wireless communications networks provides significant performance benefits but at the same time it presents also major challenges, which are attributed to the broadcast nature of wireless communications which makes the performance of these systems being dominated by the interference level.

Multicast-broadcast single-frequency network (MBSFN) is a transmission mode defined in the fourth generation (4G) cellular networking standard (LTE) to support multicast / broadcast transmission. MBSFN is based on the principles of single frequency networks (SFN), where adjacent BSs form a group sending the same signals simultaneously on the same frequency, thus allowing data to be multicast or broadcast as a multicell transmission over the synchronized SFN. The transmissions from the multiple cells are sufficiently tightly synchronized, to avoid the occurrence of inter-symbol interference (ISI). In effect, this makes the MBSFN transmission appear to user equipment as a transmission from a single large cell. MBSFN has been successfully demonstrated for mobile video distribution to mass audience in the occasion of the Super Bowl in February 2014, where live video content was transmitted over the existing 4G network infrastructure.

2.2.3 Multicast over Satellite

An apparent benefit of satellite communications over terrestrial communications is its ability to offer ubiquitous coverage and uniform accessibility. Hence, satellite communications technology facilitates inherently multicast transmissions.
2.3 Satellite Multicast

2.3.1 Benefits

Following on from the general benefits of using multicast, we highlight in this Section the benefits of employing satellite multicast transmissions.

- **Greater bandwidth efficiency**: By transmitting to multiple receivers instead of a single destination and also by eliminating the need to repeat transmissions, the available spectral resources can be efficiently utilised.

- **Uniform service provision**: Ubiquitous coverage uniform provision of services, irrespective of the location. The differences in the service level are caused only by the difference on the position of the receiving terminals within the beam coverage.

- **High capacity and scalability**: Satellite networks allow higher capacity than wired and wireless terrestrial networks. For example, Avanti’s network, which is fully capable to support satellite multicasting, has demonstrated a multicast capacity of 40Mbps downstream. Also extending the service to a new receiver / location requires minimal changes with respect to user equipment and no expensive changes in the network, while in contrast extending terrestrial communications networks is often not economically viable.

- **Highly reliable and secure transmissions (Access, Authentication, Authorisation)**: By using satellite transmission technology, we are able to
  - separate business critical data from standard IP streams;
  - guarantee delivery;
  - ensure Internet connectivity; and
  - provide enhanced reliability and up to 99% network availability.

2.3.2 NORM Protocol

According to [17], the Negative-ACKnowledgment (NACK) Oriented Reliable Multicast (NORM) protocol "can provide reliable transport of data from one or more senders to a group of receivers over an IP multicast network". Efficiency, scalability, and support for heterogeneous IP networks and for bulk transfers are said to be the goals for the protocol’s design. Another interesting target of this protocol is to provide “support for distributed multicast session participation with minimal coordination among senders and receivers”. Starting with [17], NORM is on the IETF standard track.
In [17] message types and protocol operation are explained in detail. [18] discusses goals and challenges for reliable multicast protocols in general, defines building blocks to address these goals, and gives a rationale for the development of NORM. End-to-end reliable transport of application data in NORM is based on the transmission of NACKs from the receivers to initiate repair transmissions from the senders. Variability in network conditions is taken care of by using adaptive timers for the protocol operations. The protocol is designed to offer its transport services to higher levels in a number of ways, in order to satisfy the needs of different applications.

NORM uses FEC (forward error correction) in various ways. It can use it both in the encoding of the original stream and in the repair traffic sent to the group in response to NACKs from the receivers (proactive / reactive FEC). In general, the more FEC redundancy is put in the original stream, the less NACKs will be received. Most of the potential limitations of the scalability of the protocol come from the negative feedback generated from receivers. NORM uses a probabilistic suppression of the feedback based on exponentially distributed random backoff timers. To avoid disturbing the operations of concurrent transport protocols (e.g. TCP), a congestion control scheme is specified, although alternative choices are left as a vendor-specific implementation.

Figure 2-2. Satellite multicast implementation using the NORM protocol in Avanti’s network.
2.3.2.1 NORM Building Blocks

NORM is conceptually divided in three main blocks: NORM Sender Transmission, which takes care of data transmissions and reception of feedback (NACK) messages; NORM Repair Process, which processes the feedback information and tells the first block what to retransmit; and NORM Receiver Join Policies, which relates to policies and procedures involving receivers’ admission to the data distribution. While the process of receivers joining is generally unconstrained, a sender might wish to limit the number of potential NACK senders in various ways. Other functions (congestion control, error correction etc.) are delegated to further modules.

2.3.2.2 NORM Operations

Messages in NORM are basically divided into sender messages and receiver messages: NORM_CMD, NORM_INFO, and NORM_DATA message types are generated by senders of data content, while NORM_NACK and NORM_ACK messages are generated by receivers within a session.

The NORM_DATA messages are used by senders to transmit application data and FEC encoded repair packets, while NORM_NACK messages are generated by receivers to selectively request the retransmission of missing content. NORM_CMD messages are used for various management and probing tasks, while NORM_ACK is the acknowledgement message for such commands. As it is customary in this class of protocols, the receivers schedule random backoff timeouts before sending a NORM_NACK message, which could be repeated if the hoped-for repair has not come. The sender doesn’t react to single NACK messages but rather tries to aggregate a number of them to decide how much to rewind its transmission. When it deems the rewind to be sufficient, it proceeds to the actual retransmission.

2.3.2.3 Congestion Control

Congestion control for NORM is described in [17]. It is an adaptation of the TCP-Friendly Multicast Congestion Control (TFMCC) described in [19]. It is essentially based on a rate-control approach rather than on the control of the transmission window.

2.3.3 Examples of Multicast via Satellite

2.3.3.1 UHD Video Distribution to a Network of Cinemas

One of the business examples highlighting the benefits of multicasting technology over satellite is the Ultra High Definition (UHD) film distribution to a number of cinemas at remote locations within the UK. The main characteristics of this implementation are:
Figure 2-3. Distribution of UHD films to remote cinemas in the UK.

- Simultaneous delivery of UHD video content (4K and above).
- Ka-band utilisation, i.e. there is no need for using a big satellite dish antenna.
- Demonstrated up to 40Mbps downstream and up to 300GB content files.
- Use of the NORM protocol, which implies reliable delivery, and provision of bi-directional service.
- Low-cost.
- High availability.
- High scalability.

2.3.3.2 Project iKnowledge

iKnowledge is an Avanti and UKSA (UK Space Agency) jointly funded project with the goal to develop an end to end solution for the support of education in rural and underserved areas of Tanzania, based on:
Figure 2-4. Multicast implementation in the iKnowledge project.

- Broadband Internet access via satellite.
- Distribution and management of educational and training content.
- School network monitoring and remote IT support.

Currently iKnowledge supports 250 schools in Tanzania. Multicast is employed within the project to provide:

- Educational content (eBooks, web content and media).
- Software updates.
- Live video streams of lectures.
3 Cooperation Framework that Facilitates Satellite Multicasting

3.1 Motivation

According to recent studies done by Cisco [20][21], the annual run rate for global IP traffic was 1.22ZB per year in 2016 and is forecast to reach about 3.32ZB per year by 2021. By 2021, 63% of this IP traffic is predicted to be over wireless devices, up from 49% in 2016. Also, 71% of all internet traffic will cross content delivery networks (CDN) by 2021. In terms of the contribution of video traffic to this growth, IP video traffic is predicted to grow threefold between 2016 and 2021 and contribute to 82% of the global IP traffic. Also, by 2019, online video is estimated to be responsible for four-fifths of global internet traffic. Additionally, Internet video to TV grew 50% in 2016 and is expected to increase nearly 4-fold by 2021, while consumer VoD (Video on Demand) traffic is forecast to nearly double by 2021.

Multicasting provides the capability to scale the network infrastructure at low cost while at the same achieving efficient usage of the available bandwidth, so that the increased capacity demands are satisfied, and supporting reliable data transmission, as required by several IP services.

This Section presents a process that facilitates the application of satellite multicasting and enables the optimal utilisation of the satellite link’s capacity, in order to enhance the backhaul throughput of the hybrid satellite-terrestrial network proposed within SANSA.

3.2 Satellite Capacity Utilisation Statistics

In this Section, satellite capacity utilisation statistics obtained from Avanti’s multicast-enabled VSAT network are presented. The provided data has been extracted from Avanti’s coverage over Europe over 2 beams. Capacity utilisation includes both uplink and downlink transmissions.

The figures presented here are mainly intended to provide a visual indication to how utilisation varies. Hence, actual numbers or percentages are not provided. Also, we should note that the actual bandwidth usage is dependent on the number of active terminals within the beam coverage.

Figure 3-1 shows the average bandwidth utilisation distribution over a week. Data over 4 weeks, spread over 2 months, has been considered. As it can be seen, more usage is observed towards the end of the week (with the peak noted on Tuesday being most likely attributed to some event).
Similarly, Figure 3-2 presents an indication of average hourly bandwidth usage on these beams. On the x-axis, *hour 0* represents the hour between midnight and 1am. It can be observed that, within a 24 hour period, increased usage occurs towards the evening and the earlier part of night. The average taken over the first 12 hours of a day is nearly just half of the bandwidth used in the latter half. Also, we should note that from the hourly usage distribution observed over an extended time period, we have concluded that the peak usage hours show a certain pattern.
To produce a visual statistics, data collected over a time period of 6 months is illustrated in Figure 3-3. It can be seen that usage increases towards the latter end of the evenings, with hour 23, i.e. between 11pm and 12 am, as the highest repeating peak usage hour.

### 3.3 Cooperation Framework Enabling Efficient Resources Allocation and Transmissions Scheduling for Satellite Multicast

This section outlines a process that facilitates the application of satellite multicasting at the SANSA satellite-terrestrial hybrid network. The proposed process is illustrated in Figure 3-4, capturing the information exchange and interaction between the satellite service provider (SSP) and the multicast service provider (MSP). The fundamental aim of the proposed process is to make efficient use of the free satellite capacity without affecting the usual traffic; hence it is assumed that multicasting to the network edge is performed during non-peak hours when free channel capacity is available. Further assumptions made here are:

- The MSP is usually a terrestrial operator or a media / content provider or some vertical (e.g. Apple store, Google Play etc.) who is interested in utilising a multicast service to update a number of distributed cache servers.

- Cache servers, owned and / or operated by the MSP, reside at the edge of the network.
Fig. 3-4. Schematic illustrating the proposed multicasting process.

- Decisions on multicast scheduling are made by the MSP on the basis of information provided by the SSP.

The interaction between the SSP and MSP is imperative for the decision of carrier / bandwidth allocation and multicast scheduling processes. Information, such as beam coverage, grid files – detailing the geographic coordinates (latitude/longitudes) of the footprint –, throughput, free capacity available etc. is confidential and usually exclusive to the SSP. However, these details are necessary for the MSP in the multicast decision process. For example, the SSP provides data similar
to that presented in Figure 3-1 and Figure 3-2, which is essential for the MSP to create a table with historical data on non-peak hours and non-peak days. Update on capacity utilisation data is provided hourly. Capacity utilisation data is obtained for all the beams that will be used to support multicast operation within the geographical area that the MSP intends to operate. Additionally, to support the multicast decision process, it is assumed that the MSP has information on the size of the file/data that needs to be multicast, intended geographic area (i.e. target user profile), and the due date for transmission. The MSP is also assumed to utilise the geographical coordinates available from the grid file to map the distribution of multicast group addresses. The grid information is assumed to form the basis of defining user group profiles. For example, countries/counties/cities can be defined as geographic areas within certain grid points. Based on these information, when a multicast request is identified, the MSP can create a list of options, such as suitable time-windows, to perform multicast.

4 Multicast Multi-group Precoding for the SANSA Satellite Segment with Distributed Gateways

4.1 Motivation

Having described the satellite multicasting facilitation framework in Section 3, we proceed with the study of a multicast multi-group precoding method.

More specifically, this Section focuses on the FL of the SANSA satellite segment. Satellite systems are moving from single-beam to multiple-beam architectures motivated by the SE improvement that can be achieved by aggressively re-using frequencies among different beams.

The recent DVB-S2x standard [22] and, in particular, its novel super-frame specification, has opened a new field of research related to advanced interference management techniques which facilitate the aggressive frequency re-use of High Throughput Satellite (HTS) systems.

In this context, linear joint processing techniques have shown great potential when applied to the forward link of multi-beam satellite systems [23]. However, the precoding design has to consider specific constraints imposed by practical limitations, namely the rigid framing structure of DVB-S2x and the implementation of the satellite antennas which are usually fed by a dedicated High Power Amplifier (HPA) operating close to the saturation point [24].

On one hand, the underlying framing structure proposed in DVB-S2x precludes the calculation of a precoding matrix on a receiver-by-receiver basis. This is because during one transmission period,
one frame per beam accommodates multiple receivers, each under the same FEC coding. The latter implies that each end-receiver needs to decode the whole frame in order to extract its intended data. Therefore, the DVB-S2x framing structure is particularly appealing for multi-group multicast communications, where the same data is transmitted to multiple receivers [7].

On the other hand, there exists intrinsic per-antenna power limitations on the satellite that need to be taken into account while designing the precoding scheme. While most of the works on multi-group multicast precoding have only considered SPC at the satellite [4], only few recent works have considered the individual PAC [3].

Nonetheless, all previous works in this subject are based on the assumption of an ideal feeder link, i.e. the link from the satellite GW to the satellite, with un-limited capacity. However, as stated in [4], the consideration that all spot beams in the system are served by a single GW is not practical due to the limited feeder link spectrum.

As a consequence, in this section we consider the design of multi-group multicast precoding schemes that take into account not only the DVB-S2x framing structure and the PAC, but also the feeder link bottleneck. In particular, we extend our previous works in [25][26], which are designed for terrestrial systems, to the satellite segment considering multiple GWs. In particular, to alleviate the unrealistic ideal feeder link assumption, the present work considers the clustering of beams, where each cluster is served by a single GW station.

4.2 System model

We consider a single multi-beam satellite FL transmitting to multiple hybrid BNs, which are assumed to be equipped with a satellite dish antenna and, therefore, have the possibility to receive backhaul traffic through the satellite network. To facilitate the reading, in this section, we will generally refer to these hybrid BNs simply as BNs.

Let $N_t$ denote the number of antenna elements on the satellite, which is assumed to be equal to the number of beams, and $N_i$ the number of BNs that have to be served simultaneously. The received signal at the $i$-th backhaul station can be expressed as,

$$ y_i = h_i^H x + n_i, $$

where $h_i$ is a $N_t \times 1$ vector composed of the channel coefficients between the $i$-th BN and the $N_t$ antennas of the satellite, $x$ is the $N_t \times 1$ vector of transmitted symbols and $n_i$ is the complex circular symmetric independent and identically distributed (i.i.d.) zero-mean Additive White Gaussian Noise (AWGN) measured at the $i$-th BN. Herein, for simplicity, the noise will be normalized.
to unity and the impact of the noise at the receiver side will be evaluated according to the channel coefficients, as it will be shown in the following sections.

Without loss of generality, let us assume that a number of multicast groups equal to the total number of antennas $N_t$ are realised, where $\mathcal{J} = \{ \mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{N_t} \}$ is the collection of index sets and $\mathcal{G}_k$ is the set of BNs that belong to the $k$-th multicast group, $k \in \{1, \ldots, N_t\}$. Each BN belongs to only one frame (i.e. group), thus $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset, \forall i, j \in \{1, \ldots, N_t\}$, while $\rho = N_u/N_t$ denotes the number of BNs per group (the terms frame and group will be hereafter used interchangeably).

Let $w_k \in \mathbb{C}^{N_t \times 1}$ denote the precoding weight vector applied to the transmit antennas to beamform towards the $k$-th group of BNs. By collecting all BN channels in one channel matrix, the general linear signal model in vector form reads as

$$y = Hx + n = HWs + n,$$  \hspace{1cm} (2)  

where $y \in \mathbb{C}^{N_u}$, $n \in \mathbb{C}^{N_u}$, $x \in \mathbb{C}^{N_t}$, and $H \in \mathbb{C}^{N_u \times N_t}$.

Since the FBP imposes a single precoding vector for multiple BNs, the matrix will include as many precoding vectors (i.e. columns) as the number of multicast groups (i.e. frames). This is the number of transmit antennas, since one frame per-antenna is assumed. Also, the symbol vector includes a single equivalent symbol for each frame, i.e. $s \in \mathbb{C}^{N_t}$, inline with the multicast assumptions. Consequently, a square precoding matrix is to be designed, i.e. $W \in \mathbb{C}^{N_t \times N_t}$ such that $W = [w_1 \ w_2 \ \cdots \ w_{N_t}]$, where $w_k, k \in \{1, \ldots, N_t\}$, is the precoding vector for the $k$-th multicast group.

The assumption of independent information transmitted to different frames implies that the symbol streams $\{s_k\}_{k=1}^{N_t}$ are mutually uncorrelated. Also, the average power of the transmitted symbols is assumed to be normalized to unity. Therefore, the total power radiated from the antenna array is equal to:

$$P_{\text{tot}} = \sum_{k=1}^{N_t} w_k^H w_k = \text{Trace}(WW^H).$$  \hspace{1cm} (3)  

The power radiated by each antenna element is a linear combination of all precoders [27]:

$$P_n = \left[ \sum_{k=1}^{N_t} w_k w_k^H \right]_{nn} = [WW^H]_{nn},$$  \hspace{1cm} (4)  

where $n \in \{1, \ldots, N_t\}$ is the antenna index.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
4.2.1 Multi-Gateway System Model

In the case of a multi-beam satellite served by more than one GWs, the system model includes two different types of interference: The inter-cluster and the intra-cluster interference. Let $\mathcal{T} = \{1, \ldots, T\}$ denote the total set of transmitters. The received signal at the $i$-th BN is given by:

$$y_i = \underbrace{h_{t,i}^H w_k s_k}_{\text{desired signal}} + \sum_{t \in \mathcal{G}_t \setminus \{k\}} \underbrace{h_{t,i}^H w_i s_i}_{\text{intra-cluster interference}} + \sum_{j \in \mathcal{T} \setminus \{i\}} \sum_{m \in U_j} \underbrace{h_{j,i}^H w_m s_m + n_i}_{\text{inter-cluster interference}}$$  \hspace{1cm} (5)

where $h_{t,i} \in \mathbb{C}^{N_c \times 1}$ is the channel vector from GW $t$ to BN $i$, $N_c$ is the number of beams per cluster, $w_k \in \mathbb{C}^{N_c \times 1}$ is the transmit beamforming (BF) vector of the $k$-th group, in which the $i$-th BN belongs to and thus receives the normalized symbol $s_k \in \mathbb{C}$. As illustrated in (5), the first term is the desired signal, whereas the second and third terms denote the intra-cluster and inter-cluster interference, respectively.

4.2.2 Multi-Beam Satellite Channel

We assume the multi-beam satellite beam pattern depicted in Figure 4-1, which is a 245 beam pattern that covers Europe (extracted from [28]). For the purpose of the present work, only a subset of 49 beams will be considered, which are highlighted in different colours in Figure 4-1.
A complex channel matrix that models the link budget of each BN as well as the phase rotations induced by the signal propagation is employed [29]. In more detail, the total channel matrix $H \in \mathbb{C}^{N_u \times N_t}$ is generated as

$$H = \Phi B,$$

which includes the multi-beam antenna pattern $B$ and the signal phase due to different propagation paths among the different BNs $\Phi$. In particular, matrix $B$ models the satellite antenna pattern, the path loss, the receive antenna gain and the noise power. Its $(i,j)$ entry reads as [28]:

$$b_{i,j} = \left( \frac{\sqrt{G_R G_{ij}}}{4\pi(d_k \lambda^{-1}) \sqrt{\kappa T_{cs} B_u}} \right)$$

where $d_k$ is the distance between the $i$-th BN and the satellite (slant-range), $\lambda$ is the wavelength, $\kappa$ is the Boltzmann constant, $T_{cs}$ is the clear sky noise temperature of the receiver, $B_u$ the is backhaul link bandwidth, $G_R$ is the BN antenna gain and $G_{ij}$ is the multi-beam antenna gain between the $i$-th BN and the $j$-th on-board antenna. Note that the beam gain for each satellite-BN pair depends on the antenna pattern and on the BN position.

An inherent characteristic of the multi-beam satellite channel is the high correlation of the signals at the satellite side. Thus, a common assumption in multi-beam channel model is that each receiver on the Earth will experience the same phase between all transmit antennas due to the long propagation path [29]. The identical phase assumption between one BN and all satellite antennas is supported by the relatively small distances between the transmit antennas compared with the long propagation distance of all signals to a specific receiver. Hence, in (6) the diagonal square matrix $\Phi$ is generated as $[\Phi]_{xx} = e^{j\phi_x}$, $x = 1, \ldots, N_u$, where $\phi_x$ is a uniform random variable in $[2\pi, 0)$ and $[\Phi]_{xy} = 0$, $\forall x \neq y$.

### 4.3 Proposed Multicast Multi-Group Precoding with Distributed Gateways

As mentioned in the introduction, this Section focuses on the multicast multi-group precoding design taking into account the feeder link capacity limits. As a solution, we consider here the deployment of multiple GWs to feed the satellite, where the GWs only share CSI but not the data. The same problem assuming CSI and data fully available at all GWs was studied in [30]. The latter is identical to the single GW scenario, and therefore, will be considered as an upper bound of the method presented here.
The main issue in the considered non-cooperative multi-GW scenarios is that interference from neighbouring clusters cannot be controlled, thus leading to the possible degradation of system performance, especially at the cluster edge areas. In order to mitigate these effects, GWs are allowed to coordinate when designing their precoding parameters. Coordination refers to limiting the synchronization requirements between distributed GWs that are serving the satellite. Consequently, each BN is served only by its corresponding GW.

In the multi-GW system considered herein, the interference can be controlled since CSI sharing is allowed between GWs prior to designing the respective, per-cluster precoding matrices. This coordination is assumed to be implemented centrally via the HNM. A schematic view of the considered scenario is depicted in Figure 4-2.

The system optimization objective is to maximise the minimum SINR of the $i$-th BN while satisfying the PAC at the transmitter. The optimization problem can be formulated as follows:

\[ \text{maximize} \quad \min_{i} \text{SINR}_i \]

\[ \text{subject to} \quad \text{PAC at the transmitter} \]

---

**Figure 4-2. Schematic view of the multi-group multicast FL with multiple GWs.**

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
\[
\mathcal{F}_c: \max_{s \{w_k\}_{k=1}^{N_t}} s
\]
\[
s.t. \quad \frac{1}{\gamma_i \sum_m |w_k^l h_m|^2} + \sigma_i^2 \geq s, \\
\forall i \in G_k, k \in \{1, \ldots, N_t\}, m \in \mathcal{T} \setminus \{t\}, l \in G_m \setminus \{g\}
\]
\[
\left[ \sum_{k=1}^{N_t} w_k^l w_k^l \right]_{nn} \leq P_{n,t}, \forall n \in \{1, \ldots, N_t\}, \forall t \in \mathcal{T}
\]

with \( s \in \mathbb{R}^+ \). The problem is formulated such that each satellite backhaul link’s SINR has to be equal or greater to \( s \). Thus, by maximising the minimum SINR requirement \( s \), the common SINR is maximised.

Problem \( \mathcal{F}_c \) can be solved by the well-known iterative bisection method, where the variable \( s \) gets closer to optimal value iteration by iteration. At each iteration, the feasibility of the sum power minimization problem needs to be checked for a fixed SINR requirement.

The bisection method, which solves the proposed fairness-based precoding problem in a centralized manner, is summarised in the Algorithm 1 below. This optimization procedure is performed at each GW, for all GWs in parallel. What is required is the knowledge of global CSI at each GW. This can be obtained by exchanging per-GW local CSI among all the coordinated GWs via the HNM.

**Algorithm 1 Max-min SINR**

1. Initialize \( s_{\text{min}} \) and \( s_{\text{max}} \) to be relevant minimum and maximum SINR requirements. The desired accuracy is defined by \( \epsilon \).
2. Fix the SINR requirement by solving \( s = (s_{\text{max}} + s_{\text{min}})/2 \)
3. Check the feasibility of the coordinate sum power minimization problem for a fixed \( s \). If feasible, set \( s_{\text{min}} = s \). Otherwise, set \( s_{\text{max}} = s \).
4. If \( (s_{\text{max}} - s_{\text{min}}) > \epsilon \) then go to Step 2. Otherwise, stop.
5. If the obtained solution is not rank-1, apply Gaussian randomization method (with bisection) to achieve sub-optimal, but feasible rank-1 solution.
4.4 Numerical Results

The performance in terms of minimum attainable link SINR over the coverage of a full frequency reuse, 49 beam, broadband satellite that employs FBP and is served by multiple GWs is examined in the present section. We assume that each GW is serving a cluster of 7 beams, so there are 7 distributed GWs. The multi-beam satellite pattern is the one depicted in Figure 4-1. The channel is modelled according to section 4.2.2.

Firstly, the performance degradation when no data is shared between the GWs is investigated. For accurate averaging, 100 BNs per beam are considered uniformly distributed across the coverage area highlighted in Figure 4-1. The link budget parameters considered follow the recommendations of [28] and are summarised in Table 4-1.

![Figure 4-3. Minimum SINR over the coverage area versus on-board available transmit power, for 2 backhaul nodes per frame.](image)
In Figure 4-3, the minimum satellite backhaul link SINR over the coverage of the considered multi-beam satellite is plotted versus an increasing total on-board available power. Two BNs per frame are considered, i.e. $\rho = 2$. Clearly, when a multi-GW configuration is considered, inter-cluster interference severely degrades the performance of the system and this effect increases with the transmit power. Exchanging only CSI between GWs offers a higher minimum SINR, especially in the

![CDF of backhaul link SINR over the coverage area, for 2 backhaul nodes per frame and per beam available on-board power of 50 Watts.](image)

Table 4-1. Link budget parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency Band</td>
<td>Ka-band (20 GHz)</td>
</tr>
<tr>
<td>Backhaul terminal clear sky temperature ($T_{cs}$)</td>
<td>235.3K</td>
</tr>
<tr>
<td>Satellite backhaul link bandwidth ($B_u$)</td>
<td>500 MHz</td>
</tr>
<tr>
<td>Output Back Off (OBO)</td>
<td>5 dB</td>
</tr>
<tr>
<td>Roll-off ($\alpha$)</td>
<td>0.20</td>
</tr>
<tr>
<td>Antenna Gain at receiver ($G_R$)</td>
<td>40.7 dBi</td>
</tr>
<tr>
<td>Multi-beam antenna gain ($G_u$)</td>
<td>[28]</td>
</tr>
</tbody>
</table>

In Figure 4-3, the minimum satellite backhaul link SINR over the coverage of the considered multi-beam satellite is plotted versus an increasing total on-board available power. Two BNs per frame are considered, i.e. $\rho = 2$. Clearly, when a multi-GW configuration is considered, inter-cluster interference severely degrades the performance of the system and this effect increases with the transmit power. Exchanging only CSI between GWs offers a higher minimum SINR, especially in the
lower power regime. As power increases, interference is even more challenging to mitigate. Still, considerable gains over a system without any coordination are reported.

For the same simulation setting, the cumulative distribution functions (CDFs) of the satellite backhaul link SINRs over the coverage area are given in Figure 4-4 and Figure 4-5, for different amounts of available power at the satellite. In both figures, the proposed coordinated approach offers a steeper SINR curve. This means that coordination (sharing of CSI only) reduces the variance of the SINR over the coverage, thus increasing the fairness of the system.

In the low power case, i.e. Figure 4-4, the reported SINRs are always higher than 0 dBs. On the other hand, in Figure 4-5, the high power leads to increased inter-cluster interference and therefore lower SINR values. Still, the proposed coordination between GWs manages to guarantee service availability to BNs, provided that the most robust modulation and coding parameter corresponds to at least $-5$ dB, a possible configuration according to [22]. In the same setting, the total lack of coordination translates into 97% of system availability, or in other words, 3% of the BNs over the coverage will be denied service. As expected, these nodes are situated in the cluster boarders. For satellite communications, these outage levels are considered unacceptable.

![CDF of satellite backhaul SINR over the coverage area, for 2 backhaul nodes per frame and per beam available on-board power of 200 Watts.](image)

Figure 4-5. CDF of satellite backhaul SINR over the coverage area, for 2 backhaul nodes per frame and per beam available on-board power of 200 Watts.
Furthermore, the energy efficiency of the systems under study is investigated. Firstly, in Figure 4-6, the power consumption of the coordinated system is compared to the total on board power consumed without any coordination between the GWs. As reported in [30], multi-group multicasting under PAC leads to a reduced power consumption. This is the reason why even in the single GW case, the power consumption does not reach 100%. More importantly, Figure 4-6, illustrates that the proposed coordination among GWs further reduces the total consumed power.

Next, the energy efficiency of the systems, as defined by dividing the minimum attainable SINR over the consumed power, is illustrated. For better presentation, the results are normalized over the maximum energy efficiency of the single GW system acting as a performance upper bound. As shown in Figure 4-7, energy efficiency is reduced as the total power increases. Impressively, however, in the low power regimes, the proposed coordination attains a performance close to the single GW upper bound.
4.5 Conclusions

In this work, full frequency reuse multi-beam satellites enabled by FBP are investigated. In this context, the present work focuses on the feeder link capacity limitation together with the PAC, and proposed a multi-group multicast precoding scheme for distributed GWs, where only CSI is shared among them. According to the presented results, it is shown that the interference from adjacent clusters has a significant impact on the system fairness. However, the results showed that the proposed algorithm provides good availability to the BNs in the coverage area while alleviating the feeder link transmission. Moreover, we have also shown that in low power regimes, where the inter-cluster interference is low, the proposed coordination attains an energy efficiency performance close to the single GW upper bound.

Figure 4-7. Energy efficiency versus per beam available on-board power.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
5 Caching for Hybrid Satellite-Terrestrial Backhaul Networks

5.1 Introduction

5.1.1 Mobile Data Traffic Growth

The Internet has followed an extraordinary evolution during the past four decades, thanks to several technological advances and theoretical breakthroughs that unlocked unprecedented possibilities. As a consequence, it constitutes today an integral part of the modern society, affecting and even defining in some cases fundamental aspects of our daily lives (e.g. communication, business, media, education, entertainment etc.).

However, this evolution procedure has not reached an end yet; instead, it has kept momentum. Its manifestation includes the introduction of novel services and the distribution of new types of content through this global communication network.

The transition of the Internet services to the mobile radio communications domain represents a major milestone in this timeline which, in turn, reshaped the wireless communications landscape. More specifically, the mass-production of low-cost portable terminal devices with high processing power and Internet access capabilities (e.g. smartphones, tablets, notebooks etc.), the emergence of data-hungry video streaming services (e.g. YouTube, Netflix etc.), and the increasing video quality (e.g. HD video) resulted in an enormous growth in the volume of the mobile data traffic, as reported by several measurement studies. For example, according to Ericsson, the data traffic transported through mobile radio communications networks grew about 70% between the Q1 2016 and Q1 2017 [31].

This trend is expected to continue in an even more abrupt pace in the 5th Generation (5G) era. For instance, a traffic forecast conducted by Cisco for the period 2016-2021 indicates a sevenfold increase of the overall mobile data traffic in this time interval, with mobile video traffic representing about the 77.5% of the global mobile data traffic by 2022 [2]. The anticipated traffic growth is mainly attributed to the characteristics of the envisaged services and use cases. Examples include the delivery of UHD video streams to the subscribers and the support of extreme mobile broadband (eMBB) access at crowded areas (e.g. airports, train stations, shopping malls, stadiums etc.) [32].

This exponential increase of the mobile data traffic places a heavy burden not only on the radio access network (RAN) but also on the MBH segment and the core network (CN).
5.1.2 Caching: Concept and Benefits

Caching has been identified as a means to deal with the implications of the traffic explosion. A caching system, such as a CDN run by a network operator or a CDN provider, is comprised by a number of nodes (servers) placed between the origin servers and the end users. Typically, these caching nodes are installed at the proximity of the users (i.e. at the so-called edge of the network) and each one of them is associated with a group of users\(^1\). The caches exploit the redundancy in user requests (i.e. the fact that the same objects\(^2\) are requested over and over), which is more prominent in the pattern of aggregated requests initiated by large user populations due to the mutual interests of the users, by storing popular (i.e. frequently requested) content, so that future user requests are served rather locally than by the remote origin servers.

As a consequence, the data transport paths are shortened, the traffic / bandwidth consumption in the backhaul and core networks is reduced, and the content servers are off-loaded.

These facts result in enhanced reliability (decreased number of packet losses and retransmissions) due to the avoidance of congestion / bottlenecks, improved QoS expressed as lower delays and higher throughput, and cost savings for the network operators and the content providers.

In addition, such distributed caching systems enhance the service availability through the replication of content to multiple nodes.

5.1.3 Caching Efficiency

Since the cache storage space is limited, each caching node can store only a fraction of the available content catalogue. When the requested content is found in the cache, we have a cache hit; otherwise, we have a cache miss. The main performance metric of a cache is its hit rate (or ratio), i.e. the fraction of user requests that are served by the caching node. For a caching system comprised by multiple caches, we use the average value over all cache hit rates as a performance metric. This performance measure is extensively applied in practice, due to the fact that it provides an indirect quantification of all the benefits of caching (e.g. bandwidth and cost savings, delay reduction, server offloading).

However, we should mention that one can define an arbitrary cost function that weights specific parameters of interest as desired (e.g. delay, link or server load, object size etc.) as a performance

\(^1\) For instance, in a cellular mobile radio communications network the caching servers are usually collocated with the BS.

\(^2\) In this text, we define as object a piece of cacheable content (e.g. a video chunk, an image, a file etc.).
metric. For example, the network operator might want to avoid the caching of objects at nodes with high workload or reduce the traffic over transcontinental links.

5.1.4 Reactive vs. Proactive Caching

There exist two caching paradigms. In proactive caching (sometimes called prefetching), content is transferred to the caching nodes from the origin servers a priori (i.e. before any explicit user request is made). This approach results in a semi-static cache storage that is updated at regular intervals. Typically, the transport of data to the caches takes place during off-peak hours (e.g. at night). Proactive caching is commonly used for the efficient distribution of pre-recorded IP-TV and Over-The-Top (OTT) Internet TV programs (e.g. Netflix).

In reactive caching, on the other hand, content is cached only when it is requested. This approach leads to a dynamic cache storage that is updated at the user requests’ rate. Reactive caching is typically utilised for the efficient delivery of videos stored in hosting platforms (e.g. YouTube) or embedded in web pages and online social networks (e.g. Facebook).

5.1.5 Caching Schemes

In view of the finite storage capacity of the caching nodes, it is of extreme importance to use the available storage space as efficiently as possible.

The operation of a cache is governed by the applied caching policy, which determines which objects should enter the cache and which ones should be evicted from it, so that the performance of the caching system is optimised, according to some selected metric. As we have already mentioned, usually the goal of the employed caching scheme is to maximise the cache hit rate.

To this end, the applied caching strategy predicts future user requests, typically through analysis of past requests’ logs that provide insight regarding the user behaviour (possibly in conjunction with the extraction of similarities between different objects), in order to determine the content that should be stored in the cache at each update event (be a periodic update or a user-triggered one, depending on whether proactive or reactive caching is employed, respectively).

5.1.6 Traffic Model

5.1.6.1 Independent Reference Model

Consider a catalogue of \( N \) objects \( O = \{ o_1, ..., o_N \} \), which represents all cacheable data requested by a large user population over a time period of interest, listed in order of decreasing popularity rank \( R_1, ..., R_N \), where \( R_i = i, i = 1, ..., N \).
The independent reference model (IRM) assumes that the stream of requests to these objects corresponds to a sequence of i.i.d. random variables, so that a request refers to an object $o_n$ with a constant probability $p(n)$, where $p_1 \geq \cdots \geq p_n$ [33].

5.1.6.2 Relevance of the Zipf’s Law in User Requests Patterns

Several measurements of the patterns of requests for content on the Internet revealed that the user behaviour is governed by the Zipf’s law, so that a small subset of popular objects attracts the main portion of the user requests. In particular, a Zipf distribution associated with a finite set of $N$ objects attributes decreasing request probabilities $p(n)$ to these objects corresponding to their popularity rank $R_n \in \{1, \ldots, N\}$, according to the following relation:

$$p(n) = \frac{\Omega R_n^{-\alpha}}{\sum_{n=1}^{N} R_n^{-\alpha}}, \quad \Omega, \alpha > 0$$

(9)

$$\sum_{n=1}^{N} p(n) = 1 \Rightarrow \Omega = p(1) = \frac{1}{\sum_{n=1}^{N} R_n^{-\alpha}}$$

where $\Omega$ is a normalization constant used to make the sum of access probabilities equal to one, so that (9) represents a valid probability mass function (pmf), and $\alpha$ is a shaping factor that determines the skewness of the distribution (i.e. the higher the value of $\alpha$, the longer the tail of this “power-law-like” ranking distribution). Many case studies for content requests on various platforms (e.g. web site pages [34], YouTube videos [35], IP-TV channels [36][37], files in P2P systems like BitTorrent and Gnutella [38][39] etc.) have shown that the parameter $\alpha$ typically varies in the range $0.5 \leq \alpha \leq 1$. However, some simulation studies consider also the $\alpha < 0.5$ use cases.

The optimum hit rate, under IRM conditions, of a cache with storage capacity sufficient to hold $M \ll N$ objects (an assumption that holds true throughout this Section) equals the sum of the access probabilities of the top $M$ objects, in terms of popularity ranking:

$$h_{opt}^{(M|N)} = \sum_{n=1}^{M} p(n)$$

(10)

We note that the high concentration of user requests to a small number of popular objects, as a consequence of the relevance of the Zipf’s law in Internet content access statistics, implies that even relatively small caches can be quite efficient, in terms of the achieved cache hit rate, provided that the applied caching strategy stores the most popular objects in the cache. For example, it is easy to verify using (10) for a scenario with $\Omega = 1$ and $\alpha = 0.8$ that if we cache the top-10 objects, in terms
of popularity rank, from a catalogue of $N = 1,000$ available objects, we can already achieve a hit rate of $h_{\text{opt}}^{(10(1,000))} \approx 23\%$.

5.1.6.3 Content Popularity Dynamics and Validity of the IRM

In practice, the popularity of the objects changes over time and new objects may enter the catalogue. The popularity evolution of each object is characterized by a rapid growth towards maximum popularity, especially for young objects with low initial popularity rank, followed by a phase of slow decrease [39].

In principle, unpredicted changes in content popularity could degrade the caching efficiency, if such dynamics is high enough to render the majority of the content stored in the caching node useless only a few requests after it has been loaded into the cache.

However, several studies have shown that the temporal dynamics in object popularity are noticed on the timescales of days, weeks, or months. For example, a study based on YouTube video traces concludes that we should ignore the timescale of a few hours and focus instead on the one of a few days / weeks in caching decisions [40].

Moreover, studies on P2P and IP-TV systems indicated that content popularity dynamics is relatively low. For instance, in [39] has been observed only a 1-3% daily drift in the popularity of the top 100, 1,000, and 10,000 Gnutella files. Similarly, the measurement study [37] reports that the cosine similarity value between the popularity of individual IP-TV channels over a time period of 3 days is about 0.97, corresponding to a fairly stable ranking of the content.

Thus, (9) represents a valid and yet simple approximation of the request frequency associated with objects that are characterised by slowly varying popularity.

We conclude that we can approach the optimum hit rate in practice if we hold in the cache the most popular objects over long timeframes. On the other hand, a practical caching scheme should be able to react to the popularity changes observed in realistic scenarios by replacing formerly popular cached objects with new ones that are “hot” in a recent timeframe, in order to avoid the pollution of the cache storage by outdated objects.

5.1.7 Effect of Object Size on Caching Efficiency

In this work, we consider unit-size objects (or equal-size objects in general) and measure the capacity of a cache according to the number of such objects that can be stored at it. The reason for this is threefold:
1. Object popularity is the main criterion for achieving the optimum hit rate, as we have already seen.

2. The caching of an object with size, let’s say, of \( m \) MBs can be broken down into the caching of \( m \) objects with size of 1 MB and the same popularity rank as the original object.

3. In practice, codecs and transport protocols divide videos and large files into small chunks with size of few MBs while cache servers are commonly equipped nowadays with a storage of a few TBs, thus turning the bin-packing effect into a negligible issue.

Note that we use the term “object” in the most general sense, e.g. as a representation of either a video file or a video chunk. Thus, we assume that the Zipf’s law is applied at file or chunk level, depending on the case. This is very useful, since in reality the first few chunks of a segmented video may be viewed more often than the remaining chunks and, thus, may have higher popularity.

### 5.2 Hybrid Satellite-Terrestrial Off-Line Caching

#### 5.2.1 Motivation

Caching [41][42] has been suggested as a promising solution to the tremendous traffic growth issue, particularly as a key technology for 5G networks [43], to bring the content closer to the users so that the core and access networks are off-loaded and the contents are delivered with less delay. In this direction, the works of [44][45] investigate the users’ service quality in terms of outage probability and delay for uniform channels. These works are extended to non-uniform channels in [46]. Caching is a promising approach to reduce the traffic load and can be more effective if the caching and physical layer (PHY) are designed together. PHY precoding by considering the cache content availability is studied in [47][48][49]. On top of storing the popular content in the cache, which is referred to as local gain, network coding can be used to reduce the network traffic load and get a global gain [50].

In addition, satellite systems have the ability to provide wideband backhaul links and to operate in multi/broad-cast modes for immense area coverage. Latest advances such as medium earth orbit constellations (e.g. O3B) or the planned medium earth orbit Mega-constellations (e.g. LeoSat, OneWeb) can provide more intricate ways of backhauling due to their dynamic topologies. The terrestrial backhaul is a multi-hop unicast network, hence, the cached content has to go through multiple links and has to be transmitted individually towards each BN. On the other hand, the satellite backhaul can use its wide area coverage and broadcast content to all BNs or multi-cast contents to multiple groups of BNs. The application of satellite communications in feeding several network caches at the same time using broad/multi-cast is investigated in [8][51]. The work of [51]
proposes using the broad/multi-cast ability of the satellite to send the requested contents to the caches located at the user side. Online satellite-assisted caching is studied in [8]. In this work, satellite broadcast is used to help placing the files in the caches located in the proxy servers. Each server uses the local and global file popularity to update the cache. Pushing content to the caches using hybrid satellite-terrestrial network is investigated in [52].

As we see, satellite-enabled caching is a promising 5G technique to off-load the terrestrial network. Also, the satellite communications offers a vast area coverage with high transmission rate. Hence, bringing these two technologies together can further off-load the network. The main direction is to combine the satellite and terrestrial telecommunication systems in order to create a hybrid federated CDN, which can improve the cache-feeding performance and user experience.

In this section, we propose a hybrid satellite-terrestrial backhaul network along with a novel off-line proactive caching approach to off-load the backhaul traffic of the terrestrial network. We introduce a transition matrix in order to generate local content popularity distributions. We propose using the multi-group multi-casting ability of multi-beam satellites [3] to place the contents in caches of BNs where the placed contents in the cache of a specific BN match more the most popular content of the local popularity of that BN compared to the case when mono-beam satellite is used for content placement. We use the cache hit rate as the performance metric to evaluate the selection efficiency of cached content. We measure the cache hit rate for mono/multi-beam hybrid satellite-terrestrial backhaul networks with respect to cache memory size. We also quantify the cache hit rate of the hybrid satellite-terrestrial backhaul scheme for different content popularity distributions and compare it versus the benchmark schemes.

Apart from the cache rate, we investigate the required time for content placement for the proposed multi-beam hybrid satellite-terrestrial network and compare it with the benchmark schemes. In addition, we investigate the multi-group multi-casting approach of the hybrid satellite-terrestrial backhaul network for low-density rural and high-density urban areas in terms of required time for content placement.

5.2.2 Hybrid Multi-beam Satellite-Terrestrial SANSA Backhaul Network and the Mono-beam Satellite-Terrestrial Benchmark Scheme

In this section, we introduce the hybrid satellite-terrestrial backhaul architecture that incorporates off-line caching. We propose telecommunication CDN architectures that include mono/multi-beam satellites and terrestrial components, as shown in Figure 5-1 and Figure 5-2. Such architectures can benefit from the wide area coverage of satellite broad/multi-casting. In these architectures, each edge BN is equipped with a limited memory cache. In addition to the terrestrial backhaul fibre, the edge BNs are equipped with two interfaces to receive the content, one for terrestrial backhauling
and the other for satellite backhauling. In these architectures, both satellite and terrestrial networks are used for placing the files in the caches of each cache-enabled edge BN. The mono-beam architecture uses wide area broadcast based on global content popularity for content placement in the caches. The multi-beam architecture places the content in the caches according to the content popularity of each beam, which results in more accurate content placement compared with the architecture with mono-beam satellite.

In this part, we describe the suggested caching algorithms for the proposed hybrid satellite-terrestrial backhaul architecture. Here, we focus on off-line, non-real time, hybrid satellite-terrestrial backhaul caching, which is comprised of content placement and content delivery phases. The content placement is carried out in off-peak hours and is divided into two rounds where the satellite broadcast and the terrestrial unicast are used for content placement in the first and second round, respectively. In the first round, the satellite performs the content placement based on the sole global content popularity, for the mono-beam satellite case, or the per-beam global content popularities, for the multi-beam satellite case. In the second round, the terrestrial network accomplishes the content placement based on local content popularity. In the proposed off-line caching algorithm, the cached content remain intact during the content delivery period and are updated during the next content placement period. For example, putting videos of YouTube or Netflix on the caches in advance during the night, i.e. content placement, and serving the user requests through the cached video files or the terrestrial backhaul during the day, i.e. content delivery.

In this work, we consider three different caching schemes as illustrated in Figure 5-3. The hybrid satellite-terrestrial backhaul scheme is the proposed scheme; which will be compared with the terrestrial-only and satellite-only benchmark schemes. We assume that there exists a file library with arbitrary amount of files and the users’ request falls within this file library. Furthermore, it is presumed that the backhaul edge caches keep track of the requested content by the users. The recorded requests by each cache corresponds to the local content popularity of the corresponding edge BN. The local content popularities are transmitted to the satellite GW to calculate the global content popularity.

In the mono-beam architecture of Figure 5-1, the global file popularity is derived by averaging over all the local content popularities. On the other hand, in the multi-beam architecture of Figure 5-2, the global content popularity for each beam is derived using the local content popularities of the edge BN located in the corresponding beam. In the hybrid satellite-terrestrial scheme of Figure 5-3, each cache is filled with the most local popular related files till the threshold, then the rest of the cache is filled with the most global popular files. In the case of multi-beam architecture, the global popularity of a beam is used for content placement in the caches of the BNs covered by that beam.
Figure 5-1. Hybrid satellite-terrestrial backhaul network using a mono-beam satellite.
Figure 5-2. Hybrid satellite-terrestrial backhaul network using a multi-beam satellite for more accurate content placement.
Figure 5.3. Schematic of the proposed hybrid satellite-terrestrial backhaul network and the benchmark caching algorithms.

<table>
<thead>
<tr>
<th>Benchmark 1: Terrestrial only</th>
<th>Filled based on local popularity (L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benchmark 2: Satellite only</td>
<td>Filled based on global popularity (G)</td>
</tr>
<tr>
<td>Hybrid satellite-terrestrial</td>
<td>L</td>
</tr>
</tbody>
</table>

Note that the files placed based on the local popularity are removed from the global content popularity profile(s) before content placement based on global popularity.

In the terrestrial-only method of Figure 5.3, the files are placed in the caches based on local content popularities that results in the highest cache hit rate. Hence, we use it as the upper bound to evaluate the cache hit rate of the hybrid mono/multi-beam architecture. On the other hand, in the satellite-only method, the content placement is carried out only based on global content popularity profile(s), which results in the lowest cache hit rate, hence, it is used as a lower bound for the cache hit rate.

Since the multi-beam hybrid satellite-terrestrial architecture considers the global popularity of each beam for the content placement phase, it results in placing the content in the edge caches that are more similar to the most popular files of the local content popularity of the corresponding edge BNs. Therefore, using a multi-beam satellite for the content placement phase results in a higher cache hit rate compared to using a mono-beam satellite.

In the following section, we describe the content popularity model.

5.2.3 Content Popularity model and local popularity profile generation

In this section, we define the content popularity distribution model and the proposed approach to generate the local content popularity distribution. To generate the local and global content popularities, first, we define the model for a reference content popularity distribution. Here, we assume that the reference content popularity follows the Zipf-like distribution as [34]:

\[
p(i) = \frac{\Omega}{i^\alpha}
\]

where \( \Omega = \left( \sum_{i=1}^{N} \frac{1}{i^\alpha} \right)^{-1} \) is the normalization factor to have \( \sum_{i=1}^{N} p(i) = 1 \). The local content popularity generation process for each BN is as follows. Assume that \( p \) is an \( N \times 1 \) vector which

\[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network\]

Date: 28/07/2017
accommodates the elements of the reference popularity $p(i)$ introduced in (11) for $i = 1, \ldots, N$. Let us assume that this is the first local content popularity distribution that we have generated. To generate the second local content popularity, we swap two elements of the vector $p$. To swap these two elements of $p$, first, we randomly select an element of $p$, which is a file with a specific popularity. Second, we use the probability matrix $P$ defined in (12) to find the second file which is going to be swapped with the first file.

$$P = \begin{bmatrix}
0 & p_{1,2} & \cdots & p_{1,N} \\
p_{1,2} & 0 & \cdots & p_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
p_{m,2} & \cdots & 0 & p_{m,N} \\
p_{N,1} & p_{N,2} & \cdots & 0
\end{bmatrix} \quad (12)$$

The $m$-th row of the matrix $P$ shows the probability of swapping the $m$-th file (the first randomly selected file) with other files (the second file). In the defined swapping matrix of (12), $P$, the entries closest to the diagonal entries have the highest value and then these values decrease as we move away toward further off-diagonal entries. This decrement can have arbitrary intensity. In other words, given the $m$-th and $l$-th element of $P$, $p_{m,l}$ becomes lower as $|m - l|$ increases. Therefore, the files with closer popularity have a higher chance to be swapped for local content popularity generation. This defined structure for matrix $P$ is less likely to generate consecutive local content popularity distributions that have drastic difference in content popularity. This results in correlation between the consecutive generated local content popularities since the swapped files are more likely to be closer to each other.

To generate the next local content popularity distribution, we repeat a similar procedure as before with the difference that we exclude all previously swapped files from the files, which are going to be swapped. In total, we can generate $N/2$ possible local popularity distributions by swapping the elements of $p(i)$ in (11) using the transition matrix defined in (12).

### 5.2.4 Numerical results

In this section, we simulate the performance of the proposed off-line caching algorithm in the proposed multi-beam hybrid satellite-terrestrial architecture and compare this performance with the mono-beam hybrid satellite-terrestrial, satellite-only, and terrestrial-only benchmark architectures. We measure the cache hit rate and the required time for content placement to
evaluate the proposed off-line caching algorithm in the multi-beam architecture and compare this performance versus that of benchmark architectures.

Here, we assume that half of the cache size of a BN is filled with the most popular files based on the local content popularity distribution of that BN and the other half of the cache is filled with the most popular content based on the global content popularity distribution. In all simulations, we consider a library composed of $10^4$ files. In the benchmark mono-beam hybrid satellite-terrestrial architecture, the content placement for all BNs are carried out using broadcast transmission.

In the simulation setup of the multi-beam architecture, we assume that there are 20 BNs in two different geographical regions that are going to receive the files in the content placement phase. The BNs in a specific region have high correlation in content popularity, however, each BN of one region has lower correlation with a BN located in another region. We consider 18 BNs in the first region and 2 BNs in the second region. For the multi-beam hybrid satellite-terrestrial architecture, we assume that each region is covered by a specific beam of the satellite and the content placement for the BNs covered by a beam is carried through multi-group multi-casting scheme. It is assumed that all files have the same size equal to 30 Mbyte. The values for the terrestrial backhaul rate are extracted from [53].

In the first scenario, we evaluate the cache hit rate and the required time for file placement with respect to the cache memory size. These simulations are carried out for the proposed multi-beam hybrid satellite-terrestrial backhaul architecture and the benchmark architectures when using the proposed off-line caching algorithm. As the first result, we study the cache hit rate and the required time for content placement of the benchmark schemes. The average cache hit rate for the BNs in the second region with respect to the cache memory size is presented in Figure 5-4 and the required file placement time for all BNs with respect to the cache memory size is presented in Figure 5-5.

As we see, the cache hit rate of the mono-beam hybrid satellite-terrestrial benchmark architecture is closer to the terrestrial-only benchmark scheme compared to the satellite-only for specific ranges of the cache memory. For example, for a cache memory with the capacity of 2010 files, the cache hit rate of the hybrid satellite-terrestrial method is 0.04 lower than that of the terrestrial-only method while the cache hit rate of the satellite-only method is 0.1 lower than that of the terrestrial-only scheme.

On the other hand, we see that the hybrid satellite-terrestrial method considerably cuts on the required file placement time, similarly to the satellite-only method, but the cache hit rate of the mono-beam hybrid approach is much closer to the terrestrial-only approach upper bound, compared to the satellite-only method.
Figure 5-4. Comparison among the cache hit rate of satellite-only, hybrid satellite-terrestrial, and terrestrial-only backhaul benchmark architectures versus the cache memory size for $\alpha = 0.7$.

Figure 5-5. Required time for content placement with respect to the cache memory size for mono-beam hybrid satellite-terrestrial, satellite-only, and terrestrial-only benchmark architectures.
This is due to the fact that the user requests around each BN are based on the local content popularities of that BN, however, the global popularity may not be an accurate representation of the local popularities. Hence, the hybrid scheme results in a higher cache hit rate compared to the satellite-only method. In addition, we observe that for relatively low or high cache memory sizes, the cache hit rate of all approaches are close.

Next, we show the average cache hit rate of the 2 BNs in the second region with respect to the $\alpha$ in Figure 5-6 when mono-beam hybrid satellite-terrestrial, satellite-only, and terrestrial-only benchmarks are used. We observe that for relatively low or high values of $\alpha$ the cache hit rate of all approaches are almost the same. For low values of $\alpha$, the files have close popularity values and for high values of $\alpha$, very few files have high popularity and the rest of the files have close and very low amount of popularity. Hence, based on the aforementioned information, the cache memory does not influence the cache hit rate for rather low and high values of $\alpha$. For values of $0.4 < \alpha < 1.6$, the difference among the three caching schemes increases as the cache memory decreases.
In the second scenario, we investigate the effect of multi-beam hybrid satellite-terrestrial architecture via multi-group multi-casting on the cache hit rate as well as the required time for content placement and compare it versus the mono-beam hybrid satellite-terrestrial benchmark architecture. In this scenario, we consider two beams of a multi-beam satellite where each beam covers a specific region. There are 18 and 2 cache-enabled BNs in the first and second considered beams, respectively. Figure 5-7 shows the average cache hit rate of the region covered by the second beam, 2 BNs, with respect to the cache memory size for multi-beam and mono-beam architectures. As we see, the multi-beam architecture can provide up to 0.05 better cache hit rate over a long range of cache memory sizes compared to the mono-beam architecture. This is due to the fact that a multi-beam satellite can perform the content placement based on the global popularity of each region. Compared to the global content popularity of a mono-beam satellite, the global content popularity of a specific beam of a multi-beam satellite is closer to the local content popularities of the BNs covered by that beam. This results in higher cache hit rate for multi-beam hybrid satellite-terrestrial architecture compared to the mono-beam hybrid satellite-terrestrial architecture.
The average cache hit rate for the region covered by the second beam with respect to $\alpha$ is presented in Figure 5-8. As we see, the multi-beam architecture outperforms the mono-beam over a long range of $\alpha$ in terms of cache hit rate. For relatively low and high values of $\alpha$, the mono-beam and multi-beam architectures have close values of cache hit rate. This can be justified similar as the explanation for Figure 5-6. We see that the cache hit rate of the multi-beam architecture reaches that of the terrestrial-only benchmark architecture in both Figure 5-7 and Figure 5-8, the cache hit rate upper-bound.

The required time for content placement for the proposed multi-beam hybrid satellite-terrestrial as well as the satellite-only and terrestrial-only benchmark schemes is presented in Figure 5-9. As the density of the BNs in a beam increases, the content placement required time of multi-group multicast approach decreases and falls below content placement required time of the conventional frequency reuse scheme. Hence, the multi-cast multi-group approach is a good option to be used in rural areas. On the other hand, in more dense areas such as urban areas, it is better to use the multi-group multi-casting approach for content placement.

Figure 5-8. Comparison of the cache hit rate between the multi-beam and the benchmark mono-beam architectures for different values of $\alpha$. 

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
In this section, we proposed multi-beam hybrid satellite-terrestrial backhaul architecture with multi-group multi-cast approach along with an off-line caching algorithm to reduce the required time for content placement and preserving the cache hit rate. We performed extensive simulations with respect to various parameters to investigate the performance of the proposed off-line caching algorithm for the proposed multi-beam architecture and compared it versus the mono-beam hybrid satellite-terrestrial, satellite-only, and terrestrial-only benchmark schemes. The results showed that the proposed hybrid satellite-terrestrial architecture is able to considerably reduce the required time for file placement while keeping the cache hit rate very close to the terrestrial-only upper bound. Based on the simulations, we figured out that the multi-beam hybrid satellite-terrestrial caching method can be beneficial for content popularity distribution with the range of $0.4 < \alpha < 1.6$. In addition, in order to have improvement in cache hit rate, the required cache memory for the proposed off-line hybrid satellite-terrestrial scheme varies depending on the content popularity distribution parameter $\alpha$. 

Figure 5-9. Required time for content placement with respect to the cache memory size for the multi-beam hybrid satellite-terrestrial (FFR and conventional), satellite-only, and terrestrial-only architectures.

5.2.5 Conclusions
It was demonstrated that the proposed multi-beam hybrid satellite-terrestrial off-line caching scheme can further improve the cache hit rate compared to the mono-beam hybrid architecture and almost reach to the upper bound of the cache hit rate. This is done by considering the global popularities of each beam which covers a smaller region compared to the mono-beam hybrid satellite-terrestrial architecture and can improve the accuracy of content placement. We also revealed that for a long range of values for $\alpha$, the multi-beam architecture outperforms the mono-beam architecture in terms of the cache hit rate. In addition, it was shown that using the multi-group multi-cast with the multi-beam architecture is beneficial for the case that the density of the BNs covered by each beam remain below a specific value.

5.3 Novel Hybrid Reactive Caching Strategy for Hybrid Satellite-Terrestrial Backhaul Networks

5.3.1 Introduction and Motivation

5.3.1.1 Hybrid Proactive-Reactive Caching

Each one of the aforementioned caching paradigms (i.e. reactive and proactive caching) has its weaknesses [9]. More specifically, reactive caching suffers from compulsory misses, i.e. the cache misses that take place when an object is requested for the first time. Also, the existence of one-timers (i.e. objects that are requested only once) limits its performance. The static nature of proactive caching, on the other hand, may lead to inefficient usage of the storage space, i.e. the cache may hold irrelevant data after some period of time.

Hence, in practice it is often followed a hybrid approach, where these caching methods are integrated. It has been shown manifold that this implementation outperforms the stand-alone proactive and reactive caching approaches (e.g. see [54]).

5.3.1.2 Hybrid Proactive-Reactive Caching in the SANSA Satellite-Terrestrial Backhaul Network

The hybrid satellite-terrestrial backhaul network considered in SANSA has been designed having in mind, among other things, the efficient distribution of popular multimedia content.

As mentioned previously, one technology that facilitates the achievement of this goal is satellite multicasting. In Section 5.2 a setup where the BNs serve also as caching nodes, in order to further off-load the terrestrial backhaul segment and reduce the delays involved in data transport, and it has been studied a hybrid satellite-terrestrial proactive caching scheme which exploits both satellite
multi-group multicast and terrestrial unicast transmissions to efficiently fill-in the caches with content at off-peak hours (e.g. overnight).

However, in practice it is typically followed a hybrid approach where both proactive and reactive caching are utilised, in order to mitigate their drawbacks and optimise the performance of the system, as we have already mentioned in Section 5.3.1.1.

Based on this remark, we study in this Section the application of a novel hybrid reactive caching scheme that complements the aforementioned proactive caching strategy. (Here, the term “hybrid” refers to the utilisation of both local and global file popularity information in the caching decisions.) The goal is to provide a holistic solution to the problem of efficient multimedia content delivery over the hybrid satellite-terrestrial backhaul network, based on satellite multi-group multicasting and hybrid proactive-reactive caching, and exploit the installed caching infrastructure as efficiently as possible.

Note that in the context of hybrid proactive-reactive caching, the cache storage should be partitioned into a semi-static part and a dynamic part. Therefore, the size of each partition has to be determined. Several heuristic approaches have been proposed in this regard [55]-[57]. In the scenario considered here, though, where the semi-static cache preserved for proactive caching is updated via satellite multi-group multicast and terrestrial unicast at regular intervals, the partition threshold can be determined by the multicasting framework described in Section 3.

Thus, in this Section, we simply assume a dynamic cache storage with capacity $M$, in accordance to the discussion in Section 5.1.

### 5.3.2 Score-Gated LRU

#### 5.3.2.1 Reactive Caching Overview

The operation of a caching node, assuming that a reactive caching scheme is utilised, is described as follows: Upon receiving a user request, the cache performs a lookup at its local storage. If the requested object is found there (cache hit), then it is transferred to the end user. Otherwise (cache miss), the cache fetches the object from the origin server, possibly stores a copy of it for future use, and then serves the user.

The applied caching policy is responsible for updating the contents of the local cache storage. Therefore, it affects directly the caching efficiency.

The cache storage has finite capacity, which eventually gets exhausted. Then, upon a cache miss, a cached object has to be evicted from the local storage in order to free up some space and allow the
requested object to enter the cache. The employed caching scheme determines whether a requested object will enter the cache or not (admission control policy) and, if so, which cached object will replace in case that the storage capacity of the caching node has been reached (replacement policy).

Typically, a caching strategy defines a score function that exploits one or more characteristics of the user behaviour, possibly in conjunction with additional parameters (e.g. object size), in order to assign corresponding values to the objects. Often, the caching algorithm inspects a backlog of previous requests to calculate the total score of each object. These scores serve as predictors of future user activity. Based on this information, the caching strategy determines which objects is worthwhile to hold in the cache storage and which should be dropped from it, under the goal of optimising some given performance metric.

In other words, when the storage space has been exhausted, the employed caching scheme replaces upon a cache miss the “least valuable” cached object with the requested object only if the latter object has at least equal score with the former one, in order to hold in the local storage the “most valuable” objects. In the common case where the goal of the caching strategy is to maximise the cache hit rate, the “most valuable” objects have a higher probability to be addressed again in the future than the “least valuable” ones.

We note that the requested object will enter the cache not only when it has greater score than the “least valuable” cached object but even when the two scores are equal. This implies that for two equal-value objects, the most relevant is the one requested more recently. This approach enables us to avoid the pollution of the cache with outdated objects.

Commonly, the cache storage is represented in software as a list or stack of objects that are ordered according to their score, with the “most valuable” object being placed on the top of the stack and the “least valuable” one being located at the bottom. The actual implementation of this sorted list depends on the attributes of the caching algorithm under study.

A reactive caching algorithm performs the following operations, as it is shown in Figure 5-10:

1. **Lookup**: Upon receiving a request, the algorithm performs a lookup operation to check whether the requested object is stored in the cache (cache hit) or not (cache miss).

2. **Update**: When a cache hit takes place, the score of the requested object is updated. Then, this cached object may be shifted by one or more positions towards the top of the stack, if required, so that a sorted list according to the scores of the cached objects is maintained.

3. **Insertion**: When a cache miss takes place and the cache storage is not full, the requested object enters the cache storage at the proper position, according to its score.
4. **Replacement**: When a cache miss takes place and the cache storage is exhausted, the “least valuable” object (i.e. the one placed at the bottom of the stack) is dropped from the cache and the requested object enters the cache at the proper position, only if the latter object has at least equal score to the former one (admission control condition).

5.3.2.2 **The LRU and LFU Caching Principles**

Least Frequently Used (LFU) represents the one end of the caching policies spectrum. This caching scheme counts the number of past requests to each object and holds in the cache the most frequently referenced objects. In this case, the cache storage is represented as a stack of objects ordered according to their request frequency, with the most frequently accessed object being placed on the top and the least frequently referenced one being placed at the bottom.

LFU achieves the optimum hit rate under IRM conditions, since the request frequency associated with an object reflects its popularity rank. However, this caching strategy is not used in practical applications, since it assumes an infinite backlog of past requests and a strictly ordered list of cached objects. These design parameters (a) face implementation constraints caused by storage capacity and processing power limitations; and (b) lead to pollution of the cache by formerly popular but
currently irrelevant objects that are maintained in the local storage over long timeframes. Yet, LFU serves as a benchmark when the IRM is considered.

Assuming a sequence of $k$ past requests $r_1, ..., r_k$ from the first $r_1$ to the most recent one $r_k$ within a considered timeframe, where $r_i = j$ denotes that the $i$-th request addresses the object $a_j \in O$ with popularity rank $R_j = j$, the LFU strategy is described by the score function

$$s^{LFU}_j(r_1, ..., r_k) = \sum_{i=1}^{k} \delta_{ij}; \quad \delta_{ij} = \begin{cases} 1, & r_i = j \\ 0, & \text{otherwise} \end{cases}$$

The other end of the caching strategies spectrum is Least Recently Used (LRU), which exploits the temporal locality noticed in access patterns (i.e. the fact that recently accessed objects might be requested again in the near future with high probability [58]) by assigning score values to the objects according to their time-of-last-access and storing in the cache the most recently referenced objects. Thus, the most recently accessed object is placed on the top of the cache stack and the least recently used one at the bottom.

We note that the requested object has always higher score in comparison with the cached objects. Thus, as we can see in Figure 5-11:

- In each cache miss, the requested object enters the cache (i.e. the admission control condition is always met).
- The requested object is always placed at the top of the stack. This fact has motivated a simple and efficient implementation in software of an LRU cache as a doubly-linked list of size $M$, which results in low, constant $O(1)$ effort per request for updating the cache contents by placing the most recently referenced object on the top of the stack.

LRU is the most widely adopted caching scheme due to its simplicity, low cache update overhead, and sufficient hit rate for many applications of practical interest (in comparison to other caching schemes of similar implementation complexity and computational load), as well as because of its ability to react to the popularity dynamics that stems from the limited effective size of the backlog. However, in view of the Zipf-like access patterns noticed in practice, this caching strategy is considered to be suboptimal in terms of the achieved caching efficiency, since it does not take into account the popularity of the requested objects\(^3\). The LRU deficits have been proven through analytical expressions, numerical simulations, and trace-based measurement studies [10]-[14].

\(^3\) In fact, object popularity is reflected indirectly in temporal locality in some extent. However, its effect in caching efficiency is very small.
LRU is obtained e.g. by selecting objects that maximise

\[ s_j^{\text{LRU}}(r_1, \ldots, r_k) = \sum_{i=1}^{k} 0.5^{k-i} \delta_{ij} \]  \hspace{1cm} (14)

### 5.3.2.3 Criteria for the Design of Caching Schemes

Based on the above, we define the following criteria for the design of a caching scheme [13]:

1. Simple implementation and low constant cache update effort per request, comparable to the one of LRU.
2. High hit rate that approaches the optimum, under IRM conditions, LFU caching efficiency.
3. Response in popularity shifts over time through the implementation of an aging mechanism.
that decreases the influence of old requests on the decision whether an object should enter or leave the cache.

4. Minimum traffic associated with the loading of objects into the cache and with the caching of “one-timers”.

5. Flexibility to optimise generalised cost functions.

5.3.2.4 The SW-LFU, GF-LFU and SG-LRU Caching Schemes

Several caching strategies that combine characteristics of the LFU and LRU schemes and even include them as special cases have been studied in the literature. For instance, in [11] are described the following caching schemes belonging to the so-called LRFU spectrum:

• Sliding Window LFU (SW-LFU), which restricts the LFU principle to request counts within a sliding window (SW) of the $W$ most recent requests; and

• Geometric Fading LFU (GF-LFU), which weights former requests by a decreasing factor $\rho^k$ for the $k$th recent request and ranks the objects according to the sum of their weights.

SW-LFU and GF-LFU are described by the following score functions:

$$s_{j}^{SW-LFU}(r_1, \ldots, r_k) = \sum_{i = \max(LR-W+1)}^{k} \delta_{ij}$$

$$s_{j}^{LRU}(r_1, \ldots, r_k) = \sum_{i = 1}^{k} \rho^{k-i} \delta_{ij}$$

Both variants include a single parameter, the window size $W$ for SW-LFU and the fading factor $\rho$ for GF-LFU, with main impact on the scores and the resulting ranking of objects according to scores. For both score functions, one extreme is an LFU ranking for $W \to \infty$ and $\rho \to 1$, respectively, i.e. an unlimited count of previous requests to each object. On the other extreme, GF-LFU converges to LRU for $\rho \leq 0.5$. Similarly, SW-LFU converges to LRU for $W \to 0$, assuming that LRU is used as a tie-breaking mechanism among objects with currently identical scores.

The operation of LRU, SW-LFU, and GF-LFU is illustrated schematically in Figure 5-12.

As it has been shown in [11], these schemes outperform significantly LRU, especially for small caches. However, despite their efficient implementation described in [11], the requirement of maintaining an ordered list of cached objects based on the object scores as they are determined by
the defined score functions leads to high computational complexity and turns these caching strategies impractical.

Figure 5-12. Comparison of LRU, SW-LFU, and GF-LFU caching schemes. We see that these caching strategies rank differently the objects, despite the fact that the user requests sequence is the same. More specifically, in LRU the object with ID = 7 has the highest rank, since it is the most recently requested object, and it is followed by the objects with ID = 5, ID = 6, and ID = 3 (the LRU object). In SW-LFU, on the other hand, the object with ID = 5 has a score equal to 3 (which could be denoted as $S(5) = 3$), since it has been requested three times in the sliding window of $W = 10$ past requests. Similarly, $S(6) = S(2) = 2$ but the object with ID = 6 has a higher rank than the object with ID = 2, since it has been requested more recently (LRU tie-breaker). Finally, $S(7) = 1$. In GF-LFU, we note that $S(5) = 0.9^1 + 0.9^5 + 0.9^7 = 1.97$. The score of the other objects is calculated in similar manner (e.g. $S(7) = 0.9^7 = 1$). (Adapted from [12].)

In [12] is proposed a workaround, where only a partially sorted list is maintained by the caching algorithm. In this variant, the requested object is inserted at the top of the stack, as in LRU. In case that an insertion / replacement or update operation has to take place, the score of the requested object is compared with the score of only one or two other cached objects (i.e. the bottom object in case of a cache miss or its direct upward and downward neighbours in the list in case of a cache hit, respectively). By successively performing such simple updates, the algorithm eventually transforms the list of partially ordered cached objects to a perfectly sorted list, according to the objects’ scores. Simulation results show that such simple cache updates accomplish to collect the most popular objects in the cache. Of course, they need greater time to get to their steady state.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
than the original caching schemes described in [11]. Moreover, yet these caching strategies are significantly more demanding, in terms of processing resources, than LRU.

Figure 5-13. Operation of an SG-LRU cache with size $M = 4$ that utilises the SW-LFU score function with a sliding window of size $W = 8$. At the left, we note that the requested object (purple) has the same score with the “least valuable” cached object (orange), that is, $S(\text{purple}) = S(\text{orange}) = 1$. Therefore, since it has been more recently requested, it replaces that cached object according to the LRU principle. At the center, where the requested object is the brown one and the “least valuable” cached object is the red one, we note that $S(\text{brown}) = 0 < S(\text{red}) = 1$. Therefore, the requested object does not replace this cached object. Instead, the latter object is placed on the top of the LRU stack, as it is shown at the right.

In [12]-[14] is described a score-gated LRU (SG-LRU) caching variant, which combines the LRU principle for performing simple cache updates with the SW-LFU or GF-LFU (or any other) scoring function that is used to perform admission control and collect the most popular objects (over a recent timeframe) in the LRU cache. More specifically, assuming that a SW-LFU scoring function is utilised, the difference of this algorithm with “pure” LRU is that the reference count of the objects is stored in a sliding window of $W$ previous requests as a side information and the requested object enters the cache or replaces the bottom object according to the LRU scheme (i.e. it is placed at the top of the stack) only if it has at least equal score to that object; otherwise, the bottom object is

---

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
placed on top of the stack, as it is shown in Figure 5-13. The results of numerical simulations in several use cases (i.e. under IRM conditions, utilising a model that incorporates popularity dynamics, or based on Wikipedia traces) revealed that SG-LRU provides a significant performance gain over LRU, especially for small caches, and even approaches the optimum under the IRM LFU cache hit rate, while presenting at the same time similar to LRU cache update effort and only slightly higher implementation complexity and storage space requirements. Moreover, this caching scheme reduces the loading rate of objects into the cache, thanks to the utilisation of admission control, and largely avoids the caching of “one-timers”, which degrades the caching efficiency, in contrast to LRU. In addition, the incorporation of an aging mechanism (SW) enables the adaptation of this caching algorithm to object popularity dynamics. Finally, this approach defines a new class of flexible caching strategies which combine LRU with arbitrary score-gate functions.

5.3.3 Hybrid SG-LRU: Adaptation of SG-LRU for a Network of Caches

So far, we have considered the application of SG-LRU on a single, stand-alone caching node. However, the considered caching system is comprised by \( C \) distributed caches (which, for convenience, we assume that they have the same storage capacity, i.e. \( M_i = M \), \( i = 1, \ldots, C \)). In this case, we can modify the SG-LRU scheme by taking into account in the caching decisions not only the local score of the relevant objects but also the global one, which is calculated through averaging of the corresponding local scores over all caches. We assume that the caching nodes are divided into disjoint clusters and the global score is calculated per cluster. Any clustering scheme, resulting in smaller or larger clusters, can be employed (e.g. we may assume, in accordance to the work presented in Section 5.2, that each spot beam defines a cluster of caching nodes). The local scores can be calculated at each caching node according to any desired cost function. In this work, we assume that all caches utilise the SW-LFU score function.

Assuming that the \( C \) caching nodes are grouped into \( K \) clusters of \( U = C / K \) caches each (where \( K \) equals the number of satellite spot beams), the hybrid score function associated with the object \( o_j \) (1 ≤ \( j \leq N \)) calculated at the \( i \)-th cache (1 ≤ \( i \leq U \)) belonging to the \( k \)-th cluster (1 ≤ \( k \leq K \)) is defined as [8]

\[
H_{ij}^{(k)} = (1 - t)L_{ij}^{(k)} + t \left( G_j^{(k)} - L_{ij}^{(k)} \right) = (1 - t)L_{ij}^{(k)} + t \sum_{i=1}^{U} \frac{L_{ij}^{(k)}}{U - 1}
\]  

(17)

where \( L_{ij}^{(k)} \) is the local score of the object \( o_j \) calculated at the \( i \)-th cache of the \( k \)-th cluster, \( G_j^{(k)} \) is the global score of that object calculated within that cluster, and \( 0 \leq t \leq 1 \) is a weight that determines the balance between the local and the global scores (i.e. for \( t = 0 \) we take into account only the local scores, while for \( t = 1 \) we consider only the global scores).
5.3.4 Numerical Simulation Results

In this Section, we evaluate through numerical simulations the performance of the Hybrid SG-LRU (H-SG-LRU) strategy described in Section 5.3.3, in terms of the achieved cache hit rate, for various parameter configurations. Moreover, we compare its performance to the one accomplished by using the “de-facto standard” LRU scheme in the same setup.

5.3.4.1 Single Cache Scenario: SG-LRU vs. LRU vs. LFU

Initially, we consider the single caching node scenario, in order to get some insight regarding the behaviour and performance of the SG-LRU caching policy. We are interested in calculating the average cache hit rate of SG-LRU achieved after $S = 1,000$ simulation runs. Also, we want to compare the performance of SG-LRU with the one of “pure” LRU as well as with that of LFU, which is the optimal caching policy under the IRM as we have already mentioned.

First, we compute this performance metric for different values of the sliding window’s size $W = 100; 1,000; 5,000; 10,000; 100,000$, for scenarios with different values of the Zipf shape factor $\alpha = 0.50; 0.75; 0.99$, as it is shown in Figure 5-14 – Figure 5-16, in order to fine-tune this parameter.$^4$

We assume an object catalogue of size $N = 1,000$, a cache with storage space capable of holding $M = 20$ objects (i.e. the cache size is the 2% of the catalogue size), and a stream of $R = 100,000$ user requests.$^5$

We notice that the simulation results agree with our expectations. More specifically, we note that for a small SW ($W = 100$), the performance of SG-LRU is only slightly better than the performance of LRU. However, as the size of the SW increases, the cache hit rate of the SG-LRU scheme grows significantly and approaches the optimum LFU cache hit rate. Also, we observe that in all scenarios, a SW with size equal to the 10% of the number of requests (i.e. $W = 10,000$ in our case) offers a good compromise between caching efficiency, computational burden and storage space requirements, since the cache hit rate of SG-LRU is already very close in this case to that of LFU and any additional increase of the SW size offers minor caching performance benefits which do not justify the additional computational complexity encountered and storage space needed.

$^4$ Note that in practice, there is commonly a training period for such algorithms that have a single tuning-parameter, where through usually a trial process is obtained a sufficient feasible value of this parameter, within some resolution.

$^5$ We should note that in all simulations we do not take into account the first 25% of the requests (i.e. the first 25,000 requests in our case) in the performance statistics, in order to exclude from our study the initial transient-phase where the cache gets filled with objects and focus instead on the steady-state phase.
Figure 5-14. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with $\alpha = 0.50$ against the corresponding LRU and LFU average cache hit rates.

Figure 5-15. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with $\alpha = 0.75$ against the corresponding LRU and LFU average cache hit rates.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
Next, we focus on the effect of the Zipf shape factor on the performance of the system. More specifically, we fix the size of the SW at \( W = 10,000 \) and we vary the Zipf shape factor \( \alpha = 0.50; 0.75; 0.99 \), as it is shown in Figure 5-17.

We notice that as the skewness of the objects’ popularity distribution increases (i.e. the value of the Zipf exponent \( \alpha \) gets larger), the caching efficiency increases as well, regardless of the applied caching scheme, as expected.

Moreover, we note that the performance gain of SG-LRU over LRU grows with the skewness of this distribution. This is because as the Zipf shape factor grows, the high rank objects become more popular and this fact affects positively the caching efficiency of SG-LRU which, in contrast to LRU, directly exploits the popularity information of the objects.

On the other hand, the performance gap between SG-LRU and LFU is greater for large values of the Zipf shape factor than for small and moderate values of this quantity, implying that in this case it is required a larger SW to fully capture and exploit the popularity information related with the high rank objects.

Figure 5-16. Average cache hit rate of SG-LRU for different sizes of the SW in a scenario with \( \alpha = 0.99 \) against the corresponding LRU and LFU average cache hit rates.
Then, we fix also the Zipf shape factor to $\alpha = 0.75$ (a moderate value) and we vary the cache size $M = 10; 20; 100; 200$. We note that the caching efficiency improves as the value of $M$ increases, regardless of the applied caching scheme, as expected. This is illustrated in Figure 5-18. We notice also that the performance gain of SG-LRU over LRU is greater for small caches than for large ones. This is because when the storage space is small, the loading into the cache of the subset with the most popular objects has a more apparent effect in the caching efficiency. Similarly, the performance gap between SG-LRU and LFU is smaller for small caches, since for large caches, where we store a larger subset of popular objects in the cache, the effect of using a finite backlog with an aging mechanism instead of an infinite one (as in LFU) becomes more apparent under the IRM.

Finally, we fix the cache size at $M = 20$ and we vary the catalogue size $N = 100; 200; 1,000; 2,000$. Note that we have kept the catalogue size over cache size ratios $N/M$ the same as with the ones in Figure 5-18, i.e. $N/M = 5; 10; 50; 100$. We notice similar behaviour and caching efficiency for the same catalogue size over cache size ratio values, as expected. In other words, this test demonstrated that it is not the absolute cache size that affects the caching efficiency but it is rather the ratio of the catalogue size over the cache size that determines the performance of the applied caching scheme. This is illustrated in Figure 5-19.

Figure 5-17. Average cache hit rate of SG-LRU for different Zipf shape factor values against the corresponding LRU and LFU average cache hit rates.
Figure 5-18. Average cache hit rate of SG-LRU for different cache sizes against the corresponding LRU and LFU average cache hit rates.

Figure 5-19. Average cache hit rate of SG-LRU for different catalogue sizes against the corresponding LRU and LFU average cache hit rates.
5.3.4.2 Network of Caches: Hybrid SG-LRU vs. LRU

In this Section, we focus on the Hybrid SG-LRU variant presented in Section 5.3.3. We are interested in calculating the average cache hit rate of Hybrid SG-LRU and LRU achieved after \( S = 1,000 \) simulation runs, for a setup with \( U = 5 \) caching nodes belonging to the same cluster. We assume that each cache has a storage space equal to \( M = 20 \). Moreover, we assume an object catalogue of size \( N = 1,000 \), a SW of size \( W = 10,000 \), a Zipf shape factor equal to \( \alpha = 0.75 \), and a stream of \( R_T = 1,000,000 \) total user requests, resulting in \( R = 100,000 \) user requests at each caching node in average. We compare the performance of these caching schemes with the optimum LFU cache hit rate obtained in a setup with a single cache of the same storage capacity with these caching nodes. The segment of the first 25,000 requests is excluded from the performance evaluation.

First, we compare the performance of the H-SG-LRU scheme versus the performance of LRU for different values of the weight \( \epsilon = 0.00; 0.25; 0.50; 0.75; 1.00 \) that affects the balance of the local and the global scores in the determination of the total objects’ score. We assume different levels of similarity between the aggregated requests patterns at the caching nodes, since this quantity affects the behaviour and performance of Hybrid SG-LRU for the different values of \( \epsilon \). This level of similarity is expressed by the similarity factor \( SF \), which takes values in the range \([0.00, 1.00]\), with \( SF = 0.00 \) denoting the extreme case where the requests at different caches address different objects and \( SF = 1.00 \) denoting the other extreme scenario where the requests at different caches address the same objects. (Of course, the order with which these objects are requested by the users is different at each cache.) Besides these two extreme and highly unrealistic scenarios, we study also the realistic use case where \( SF = 0.90 \). Note that these are average values after 1,000 simulation runs, e.g. there may be \( SF = 0.92 \) in 500 simulation runs and \( SF = 0.88 \) in another 500 simulation runs, so that in average \( SF = 0.90 \).

The simulation results are illustrated in Figure 5-20, and they agree with our intuition. More specifically, we note that the Hybrid SG-LRU scheme outperforms significantly the LRU strategy. When we take into account only the local score of the objects (i.e. \( \epsilon = 0.00 \)), its performance is identical with that of SG-LRU when applied to a standalone cache. When there is no global sharing of the user interests (i.e. \( SF = 0.00 \)), the use of both the local and the global score (i.e. \( \epsilon > 0.00 \) and \( \epsilon = 1.00 \)) in the determination of the total score of the objects has no effect on the caching efficiency. Moreover, in this scenario (\( SF = 0.00 \)), when we consider only the global score in the total score calculation (i.e. \( \epsilon = 1.00 \)), the Hybrid SG-LRU strategy converges to LRU. In the other extreme scenario where \( SF = 1.00 \), the use of the global score together with the local score (i.e. \( \epsilon > 0.00 \)) provides a minor performance boosting in comparison with the use case where only the local score is utilised (i.e. \( \epsilon = 0.00 \)), similar to the one obtained when we use a slightly larger SW size, thus enabling the Hybrid SG-LRU scheme to approach closer the LFU cache hit rate. Finally, in
the scenario where $SF = 0.90$, the caching efficiency and the behaviour of the Hybrid SG-LRU scheme is very similar to the one noticed when $SF = 1.00$, since after all the Zipf-like access patterns and the 90% global sharing of user interests imply that the different user request streams differ only on the low rank objects that they address, whose impact on the performance of the system is negligible. The only difference between these two scenarios is that when in this case ($SF = 0.90$) it is utilised only global popularity information (i.e. $t = 1.00$), the caching efficiency is slightly reduced, as expected, since the global requests sharing is not 100%.

For the rest of our experiments, we choose $SF = 0.90$, as a realistic level of requests’ sharing between the different user populations / caches, and $t = 0.75^6$.

![Figure 5-20. Average cache hit rate of Hybrid SG-LRU vs. LRU in a network of $U = 5$ caches for different level of requests’ sharing between the caches and different values of the local-global popularity balance weight against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node.]

---

6 During an initial training period, we can obtain the user requests’ similarity factor $SF$ and then match the weight $t$ to this factor.

---

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
Next, we compare the performance of the Hybrid SG-LRU and LRU strategies for different values of the Zipf shape factor $\alpha = 0.50; 0.75; 0.99$. As it is shown in Figure 5-21, the Hybrid SG-LRU scheme outperforms significantly the LRU strategy, especially for small and moderate values of the Zipf shape factor. Its performance approaches the optimum LFU cache hit rate in all cases. The performance gap between the Hybrid SG-LRU and the LFU grows for large values of $\alpha$. Also, due to the use of a weight $t > 0.00$ in a scenario with $SF > 0.00$, the Hybrid SG-LRU strategy performs slightly better than the SG-LRU scheme applied in a single-node caching system, since the use of global popularity information in conjunction with the local score in the former case resembles effectively the use of a larger SW in the latter one. The only exception is the use case where $\alpha = 0.99$, in which Hybrid SG-LRU performs slightly worse than SG-LRU, as it is illustrated in Figure 5-22.

In this use case, H-SG-LRU approaches the cache hit rate of SG-LRU only when solely the local score is utilised in the determination of the objects' total score, as it is depicted in Figure 5-23, due to the fact that the high skew of the user requests’ distributions imply that the high rank objects are extremely popular. The same thing happens when we exploit solely the global score of the objects, with a negligible performance degradation, due to the fact that the level of user requests’ sharing is so high.
Figure 5-22. Average cache hit rate of Hybrid SG-LRU in a network of \( U = 5 \) caches vs. the average cache hit rate of SG-LRU applied in a single-cache system for different values of the Zipf shape factor.

Figure 5-23. Average cache hit rate of Hybrid SG-LRU in a network of \( U = 5 \) caches vs. the average cache hit rate of SG-LRU applied in a single-cache system for \( \alpha = 0.99 \) and different values of the weight \( t \).
Then, we fix the value of the Zipf shape factor at $\alpha = 0.75$ and we vary the size of the caches $M = 10; 20; 100; 200$. As we note in Figure 5-24, as the size of the caches increases, the caching efficiency increases as well, as expected. Also, as we noticed in the single-cache case, the Hybrid SG-LRU strategy outperforms significantly its LRU counterpart, especially for small to moderate cache sizes. Another similarity with the single-cache scenario is that the performance gap of the Hybrid SG-LRU scheme with the LFU strategy is smaller for smaller caches.

In the following test, we fix the cache size at $M = 20$ and we vary the number of caches $K = 2; 5; 10$. We note in Figure 5-25 that the performance difference between the various use cases is negligible, implying that it is not the number of caches that affects the average cache hit rate but it is instead the storage capacity per caching node (or, equivalently, the ratio of the catalogue size over the per-node cache size).

Finally, we study the performance of the aforementioned caching schemes for a scenario where as the number of caches varies $U = 2; 5; 10$, the cache size varies accordingly as well $M = 50; 20; 10$, so that the total size of the composite cache, $M_T = \sum_{i=1}^{U} M_i$, remains constant and equal to $M_T = $...
100. The results are depicted in Figure 5-26. In accordance with the insights obtained through the previous test case, we note that it is not the composite cache size that affects the performance of the system but it is instead the storage capacity of each caching node (or equivalently the catalogue size over per-node cache size ratio). Moreover, we notice again that for smaller caches, the performance gap between the Hybrid SG-LRU scheme and the LFU strategy is smaller than for larger caches, as expected.

5.3.5 Conclusions

In this Section, we described initially a SG-LRU caching strategy which utilises (a) an LRU cache and (b) a SW-LFU score-gate function that assigns score values to the objects based on their request frequency (popularity) in a recent timeframe defined by a sliding window and provides an admission control functionality. As a result of this approach, (a) the cache updates require a constant $O(1)$ effort per request, as in LRU; (b) the optimum, under the IRM, LFU cache hit rate is approached, since the most popular objects in the timeframe of interest are loaded into the cache; (c) the cache reacts to the object popularity dynamics noticed in practice, in contrast to LFU, since the SW represents an aging mechanism; and (d) the loading rate of objects into the cache and the storage of “one-timers” is reduced, in comparison to the conventional LRU scheme, thanks to the admission control functionality. Also, this caching strategy requires minimal additional implementation and computational complexity as well as storage space, in comparison to LRU.

Figure 5-25. Average cache hit rate of Hybrid SG-LRU vs. LRU for a system with varying number of caches of fixed cache size against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node.

Figure 5-26. Average cache hit rate of Hybrid SG-LRU vs. LRU for a system with varying number of caches of fixed cache size against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node.
The presentation of SG-LRU was followed by the evaluation of its performance in a single-cache system, based on an extensive set of numerical simulations. The simulations indicated that a SW with size equal to the 10% of the number of requests that the cache server addresses per day (or over a time frame of interest) provides a good compromise between performance on the one hand and computational complexity and storage capacity requirements on the other. Moreover, the simulation results revealed that SG-LRU outperforms significantly LRU and approaches the optimum, under the IRM, LFU cache hit rate. The performance gain of SG-LRU over LRU and the performance gap between SG-LRU and LFU becomes larger and smaller, respectively, for smaller caches. Similarly, the performance gain of SG-LRU over LRU grows with the Zipf shape factor, but the same stands also for the performance gap between SG-LRU and LFU. We should note that the term “small” or “large” cache has to do actually with the ratio of the catalogue size over the cache size instead of the absolute value of the caching node’s storage capacity, as it has been shown by the simulations.

Next, we proposed a Hybrid SG-LRU strategy that is suitable for use in a network of caches divided in disjoint clusters and can be used either as a stand-alone solution or in a complementary fashion to the proactive caching scheme described in Section 5.2. This caching scheme defines a hybrid score

\[ \text{Hybrid Score} = \frac{\text{SG-LRU Hit Rate} + \text{LFU Hit Rate}}{2} \]

Figure 5-26. Average cache hit rate of Hybrid SG-LRU vs. LRU for a system with varying number of caches of varying cache size, so that the size of the composite cache is fixed, against the average cache hit rate of LFU for a single-cache system with equivalent storage capacity per caching node.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
function that takes into account both the local and the global popularity information for each object. This function is the weighted sum of the local score and the global score of the object of interest, with the local score being the score (request count within the SW) of this object calculated at the caching node and the global score being the average of the local scores of this object calculated at the other caching nodes of this cluster.

Finally, we evaluated the performance of this caching policy through numerical simulations. We studied initially the impact of the weight that defines the balance between the local and global scores in the determination of the total score of each object for different levels of request’s sharing across the individual caches. We noticed that when there is no request’s sharing among the caches, it is only the local popularity that affects the caching efficiency. In this case, if we consider only the global scores in the hybrid score function, the performance of this caching scheme becomes identical with this of LRU, as expected. When there is some level of request’s sharing, then the caching efficiency is mainly affected again by the local popularity but this time the exploitation of the global popularity information in the calculation of the hybrid scores improves slightly the caching efficiency. The effect of using greater values of the local-global scores weight, which favour the global scores over the local ones, becomes more apparent when the level of request’s sharing is larger, as expected. Due to the minor performance boost provided by the use of global scores when there is some level of requests’ sharing at the different caches, the Hybrid SG-LRU strategy performs slightly better than the conventional SG-LRU scheme applied in standalone caching nodes. Also, as in the single-cache setup, we noticed that the Hybrid SG-LRU strategy outperforms significantly its LRU counterpart and approaches the LFU cache hit rate achieved for a standalone cache with the same storage capacity as the one of each caching node in this multi-cache network. The performance gain of the Hybrid SG-LRU scheme over LRU and its performance gap with LFU increases and decreases, respectively, for smaller caches or and for moderate Zipf shape factor values. In addition, we noticed that it is not the number of caches that affects the performance of the system but it is instead the size of each caching node (or, equivalently, the ratio of the catalogue size over the per-node cache size) that determines the caching efficiency.

5.4 (SG-)LRU-based Cooperative Reactive Caching Schemes

5.4.1 Introduction and Motivation

Cooperative caching has been recently proposed as a means to off-load the backhaul in small cell network (SCN) setups [15]. In this caching paradigm, the distributed caching nodes are typically grouped into disjoint clusters. The caches in each cluster cooperate with each other, thus forming essentially a larger composite cache.

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
Local Hit

![Diagram showing local hit in cooperative caching system](image)

Global Hit

![Diagram showing global hit in cooperative caching system](image)

Figure 5-27. Local hit vs. global hit in cooperative caching system with $U = 3$ nodes. The requests arrive at Cache #1. All nodes operate as LRU caches with size $M_i = M_0 = 4$. The caching of objects at the target cache when a global hit occurs is allowed.

More specifically: A user request addresses initially a target cache. In case of a local cache hit (i.e. the requested object is found in the target cache), this cache serves the user, while in case of a local cache miss it conducts the other caching nodes of the cluster (called remote caches in this context). If the requested content is found in one of these remote caches (global cache hit), it is transferred to the end user (possibly through the target cache); otherwise (global cache miss), it is downloaded from the origin server. This cluster-centric approach reduces the cooperation overhead and enables the scaling of the system.

5.4.2 A Family of Cooperative Reactive Caching Schemes

In this Section, we propose a family of (SG-)LRU-based collaborative reactive caching schemes. We assume that there are $U$ caches per cluster, with each one having a storage capacity equal to $M_i = M_0 = M/U$ ($i = 1, ..., U$).

We consider two distinct protocol design approaches that differ in the behaviour of the target cache in the global hit use case. That is, depending on the employed protocol, the target cache may or may not store a copy of the requested object at its local storage for future use. In the former case
(Protocol 1), a single object is stored in multiple caches, as it is illustrated in Figure 5-27, while in the latter case (Protocol 2) it is ensured that each cache holds different objects.

Of course, there is a trade-off associated with each one of these approaches, as one might expect. More specifically, when Protocol 1 is utilised, it is intuitively anticipated that the users might be served more often by the local storage of the cache associated with them (target cache) instead of a neighbouring caching node (remote cache). However, due to the smaller effective size of the composite cache caused by the overlapping in cached contents, there might be less global hits as well and, thus, the origin servers might have to be conducted more often. Similarly, in Protocol 2 the available distributed cache storage is utilised more efficiently, thus the number of global misses might be reduced, but there might be also fewer local hits, in which case more aggressive cooperation will be required. In other words, these protocols affect the local hit rate vs. global hit rate balance.

We have to note that the performance of the system depends on the characteristics of the employed caching algorithms as well as on the inner-mechanics of the utilised protocols and the given configuration. Regarding the former aspect, it is apparent that LRU and SG-LRU (the considered caching schemes in our study) have fundamentally different attributes. Therefore, substantial differences (both qualitative and quantitative) in terms of the achieved local hit rate and global hit rate are expected.

Considering the latter aspect, we should mention that we study two variants of each one of the aforementioned collaborative caching protocols, which differ in the behaviour of the remote cache under a global hit use case. In Variant A, the remote cache that transfers the requested object to the target cache does not update the contents of its local storage according to the current request (i.e. by putting the requested object on the top of the cache stack, if it is not already placed there), in order to reduce the computational load imposed on it for the purpose of helping the target cache. In Variant B, on the other hand, such a cache update takes place in this remote cache.

In addition, we consider several implementations of the SG-LRU caching scheme, which affect the operation of the remote or / and target cache upon a global cache hit. More specifically, in Type I implementation, a SG-LRU cache that is not allowed to update its contents in the scenario of interest, can still update its sliding window (SW) upon a global cache hit. This refers to the target cache in Protocol 2.B.I, the remote cache in Protocol 1.A.I, or both in Protocol 2.A.I. In Type II implementation, on the other hand, such a cache is not allowed to update the contents of its SW.

Finally, it is apparent that the operational settings, such as the value of the Zipf exponent, the number of caches, the size of each cache, the size of the SW etc. affect too the performance of the system.
It is the interplay of all the aforementioned parameters that determines at the end the behaviour and performance of the system.

In summary, in Protocol x.x.x the first digit (1 or 2) and the second one (A or B) determine whether the target cache and the remote cache, respectively, will update or not their contents upon a global hit, while the third digit (I or II) specifies whether a cache that does not update its contents under this scenario (be a target cache or a remote cache) will update its SW when the SG-LRU algorithm is utilised or not, as it is listed in Table 5-1.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Target Cache</th>
<th>Remote Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cache Update</td>
<td>SW Update</td>
</tr>
<tr>
<td>1.A.I</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1.A.II</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1.B</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2.A.I</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>2.A.II</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>2.B.I</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>2.B.II</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

One could argue that the use of “global statistics” regarding past requests in Type I implementation of SG-LRU for the purpose of updating the scores of the objects, which is an operation that involves minimal effort and storage space requirements, helps in capturing the general trend of the user behaviour and, thus, resembling better the functionality of an equivalent caching node that serves a large user population. However, a contra argument could be that this indirect “filtering” of the user requests actually distorts the Zipf distribution of object popularity, as it is “seen” by the caching node, and leads to a degradation of the performance of SG-LRU.

Note that Protocol 2.A (or 2.A.II when SG-LRU is utilised) is equivalent with a scenario where there is a single caching node shared among the users with size equal to the sum of the size of all the caches, i.e.

$$M_T = \sum_{i=1}^{U} M_i = \sum_{i=1}^{U} M_0 = UM_0 = U \left( \frac{M}{U} \right) = M \quad (18)$$

In variants 2.A and 2.A.II, any direct or indirect filtering of the user request stream through cache or SW update at the target or/and remote cache upon a global cache hit is avoided.

Our goal in this study is to investigate the effect of the aforementioned “flavours” of Protocol 1 and 2 under various configurations as well as the impact of adding more caching nodes while keeping
the effective size of the composite cache constant, in order to determine which variant suits better to each given use case. Of course, the latter test scenario seems to be meaningless when Protocol 2.A (or 2.A.II) is employed, since in this use case it is the size of the composite cache that matters, not the number of caching nodes required to reach that size. Nevertheless, it is natural to expect that in these scenarios the performance of the system will scale with the number of caches when the size of the composite cache is not upper-bounded.

5.4.3 Numerical Simulation Results

In this Section, we present the simulation results for the cooperative caching paradigm discussed in Section 5.4.2. We assume that the LRU and SG-LRU (with SW-LFU score-gate function) caching schemes are applied at the caches, in conjunction with the collaborative caching protocols presented previously. We study this caching technique under varying Zipf exponent $\alpha$, composite cache size $M$, number of caches $U$, cache size $M_0 = M/U$, and SW size $W$ (in the case where the SG-LRU scheme is employed), assuming in all cases a catalogue size $N = 1,000$. We are interested in the average local, global, and total cache hit rate obtained after $S = 10,000$ simulation runs for $R = 100,000$ requests per node in average. (The initial warm-up period is excluded from the performance evaluation, as in the previous Sections.)

More specifically, in Figure 5-28 is illustrated the performance of a system setup with a composite cache of physical size $M = 100$ objects (i.e. 10% of $N$) corresponding to $U = 5$ caches with storage of size $M_0 = M/U = 20$ objects (i.e. 2% of $N$) and varying values of the Zipf exponent $\alpha = 0.50; 0.75; 0.99$. Some interesting remarks can be made based on this Figure:

- All Protocols: As $\alpha$ increases, the total cache hit rate grows as well, as expected (especially for the case where SG-LRU is utilised).

- Protocol 1: SG-LRU exploits the relevance of Zipf's law in content access statistics to achieve a high cache hit rate. Under Protocol 1, the target cache updates both its local storage and its SW scores upon a global hit, thus distorting the popularity distribution of the objects. This is the reason why LRU performs slightly better than SG-LRU in this scenario. We notice an exception to this statement when $\alpha$ is small, since then the distortion of the objects' ranking is less important. The situation becomes worse for SG-LRU when the remote cache updates its local storage and SW as well upon a global hit (Protocol 1.B), due to the additional correlation of the user requests that is introduced into the system in this scenario. An interesting remark is that under Protocol 1.A, where the target cache updates both its local storage and SW but the remote cache does NOT update its local storage, it is preferable to allow the remote cache to update at least its SW (Type I implementation) than forbid this operation (Type II implementation), since the use of “global statistics” in the
former scenario improves the global hit rate and enhances the overall system performance. Nevertheless, it is the local hit rate that dominates in general the performance of SG-LRU when Protocol 1 is utilised, due to the fact that in this use case the target cache updates its contents upon a global hit. On the other hand, since the operation of LRU is not based on object popularity, the achieved local / global hit rate balance when this caching scheme is employed depends on the value of $\alpha$ (which, in turn, affects the efficiency of each LRU cache).

- Protocol 2: In Protocol 2, where the target cache does NOT update its contents upon a global hit, SG-LRU outperforms significantly LRU. The best performance is noticed, as expected, when the variant 2.A.II is utilised, where neither the target cache nor the remote one update their local storage or their SW, since then the SG-LRU can fully exploit the Zipf’s law to enhance the caching efficiency. On the other hand, the fact that the target cache does NOT update its local storage when a global hit takes place is responsible for the dominance of the global hit rate in the performance of both protocols, especially as $\alpha$ increases.
Figure 5-28. Average composite, local, and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. Zipf exponent for a system setup with $M = 100$ ($U = 5, M_0 = M/U = 20$), $N = 1,000$ and $W = 10,000$. (a) $\alpha = 0.50$. (b) $\alpha = 0.75$. (c) $\alpha = 0.99$. 

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]
In Figure 5-29 is depicted the performance of the system when $\alpha = 0.75$ and the composite cache size varies $M = 10; 100; 1,000$ (i.e. a composite cache having the 1%; 10%; 100% of the size of $N$, respectively), corresponding to a fixed number of $U = 5$ caching nodes with varying cache size $M_0 = M/U = 2; 20; 200$. We note the following:

(a)
Figure 5-29. Average composite, local, and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. composite cache size for a system setup with $\alpha = 0.75$, $U = 5$, $N = 1,000$ and $W = 10,000$. (a) $M = 10$ ($M_0 = 2$). (b) $M = 100$ ($M_0 = 20$). (c) $M = 1,000$ ($M_0 = 200$).

- **All Protocols**: As $M$ increases (i.e. $M/N \to 1$), the total, local, and global cache hit rate grows as well, as expected.

- **Protocol 1**: The typical behaviour of these caching schemes under Protocol 1 is observed (i.e. LRU outperforms SG-LRU and SG-LRU performs better in 1.A than in 1.B, especially when 1.A.I is utilised instead of 1.A.II). Also, the performance of LRU is dominated by the local hit rate when smaller caches are used and by the global hit rate when larger caches are utilised, while the performance of SG-LRU is determined mainly by the local hit rate. The only case where SG-LRU performs better than LRU is when the cache size is small, since in this case the exploitation of the popularity of the top-rank objects in caching decisions affects significantly the caching efficiency.

- **Protocol 2**: Again, these caching schemes perform as expected under Protocol 2. The only interesting remarks here are that LRU outperforms SG-LRU when $M = N$, a phenomenon that has to do with the selection of the SW size and the time needed for SG-LRU to converge towards an almost optimum caching scheme over the timeframe of interest; and that when the target cache does not update its local storage while the remote cache updates both its
contents and its SW (Protocol 2.B) and the cache size is small, it is better to allow the target cache to update its SW scores (Type I implementation) instead of forbidding this operation, since the use of “global statistics” improves the global hit rate and, in turn, enhances the overall system performance.

In Figure 5-30 we study the effect of varying the SW size $W = 100; 1,000; 10,000$ on the performance of the system when the SG-LRU strategy is utilised and the settings are $\alpha = 0.75$ and $M = 100$ (i.e. $U = 5$ and $M_0 = 20$). When the SW size is small, the performance of SG-LRU is mainly driven by the temporal locality and, thus, is similar to LRU. When Protocol 1 is used, where the target cache updates its contents and SW upon a global hit, the use of a large SW results in pollution of the SW with scores that do not reflect Zipf’s law, due to the filtering of the user requests. Thus, in this case, the use of an intermediate SW size is recommended. On the other hand, when Protocol 2 is used, the utilisation of a large SW is better. Of course, after some threshold value, the performance enhancement is expected to be small, as it is indicated by the trend in the simulation results, and, thus, it might not worth to use the corresponding additional storage space.

However, the most important outcome of this test case depicted in Figure 5-30 is that SG-LRU performs better than LRU under any Protocol, provided that the SW size has been selected appropriately: Protocol 1 requires the use of intermediate SW size (e.g. $W = 1,000$), while Protocol 2 benefits from the use of a larger SW (e.g. $W = 10,000$).

Under this context, one can also choose a Protocol according to his/hers needs: if the achievement of the maximum possible composite hit rate is the objective, then Protocol 2 should be utilised. However, the performance of this Protocol depends mainly on the global hit rate, which implies that Protocol 2 results in more inter-cache data transfers. If this is an issue, then Protocol 1 can be employed instead. In this case, the achieved total hit rate will be lower but the performance of the system will be dominated by the local hit rate. On the other hand, the global miss rate may be high, and this fact will lead in more often downloading of the requested objects from the corresponding origin servers.

Finally, in Figure 5-31 we keep fixed the composite cache size $M = 100$ and vary the number of caches $U = 2; 5; 10$ (and the corresponding cache sizes $M_0 = 50; 20; 10$, so that $M_0 = M/U$), for a use case where $\alpha = 0.75, W = 10,000$ and $N = 1,000$.

As we have already mentioned, in practice, it is required a training period wherein the system learns the user requests statistics in order to fine-tune the parameter $W$, so that both high hit rate is achieved and adaptation to popularity dynamics is accomplished. This procedure is very common when parametric algorithms are applied.
D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network

Date: 28/07/2017
Figure 5-30. Average composite, local, and global cache hit rate of SG-LRU for various collaborative caching schemes vs. sliding window size for a system setup with $\alpha = 0.75, M = 100, U = 5, M_0 = 20$, and $N = 1,000$. (a) $W = 100$. (b) $W = 1,000$. (c) $W = 10,000$.

We note that the performance of the system is better when fewer caches with larger storage capacity are utilised. We also notice that when the caches have large storage capacity, then it is the local hit rate that dominates the performance of the system. On the other hand, when the nodes are equipped with small caches, it is the distributed cache as a whole that determines the system performance.

5.4.4 Conclusions

In this Section, we presented various collaborative caching techniques that can be applied in a network of caching nodes divided into disjoint clusters and utilise the SG-LRU or the standard LRU caching scheme. The various flavours of these techniques differ on whether the target or remote cache will update its cache contents or its sliding window (in case that SG-LRU is used) when a global hit occurs. The numerical simulations provided useful insights regarding the behaviour and performance of this collaborative caching paradigm:
D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network

Date: 28/07/2017
Figure 5-31. Average composite, local and global cache hit rate of SG-LRU and LRU for various collaborative caching schemes vs. number of caches for a system setup with $\beta = 0.75$, $M = 100$, $N = 1,000$, and $W = 10,000$. (a) $U = 2$ ($M_0 = 50$). (b) $U = 5$ ($M_0 = 20$). (c) $U = 10$ ($M_0 = 10$).

Protocol 1: The target cache updates both its contents and its SW scores upon a global hit. When the Protocol 1 is utilised, LRU outperforms SG-LRU, due to the fact that the filtering of the user requests at the target cache distorts the ranking of the objects. An exception is when the Zipf shape factor is small, since then the distortion of the objects’ ranking is less important. In Protocol 1.B, where the remote cache too updates both its local storage and its SW upon a global hit, the performance gain of LRU over SG-LRU becomes more apparent, due to the greater distortion of the objects’ ranking. In Protocol 1.A, where the remote cache does not update its local storage upon a cache hit, SG-LRU performs better in Type I implementation where the update of the remote cache’s SW is allowed than in Type II implementation where this operation is prohibited, since in the former case it benefits from the exploitation of global statistics which enhance the global hit rate.
In Protocol 1, the performance of SG-LRU is dominated by the local hit rate, due to the fact that the target cache updates its local storage. On the other hand, when the LRU strategy is used, which exploits the temporal locality noticed in the user requests pattern, the local / global hit rate balance depends on the value of the Zipf exponent.

Also, the performance of LRU is dominated by the local hit rate when smaller caches are used and by the global hit rate when larger caches are utilised. When the cache size is small, SG-LRU outperforms LRU, due to the fact that in this case the exploitation of the popularity of the top-rank objects affects significantly the caching efficiency.

SG-LRU converges to LRU when the SW size is small, as expected. On the other hand, when Protocol 1 is used, the utilisation of a large SW results in pollution of the SW due to the filtering of the user requests caused by the fact that the target cache updates both its contents and its SW upon a global hit. Thus, an intermediate SW size is recommended in this scenario.

**Protocol 2: The target cache does NOT update its contents upon a global hit.**

When the Protocol 2 is used, SG-LRU outperforms significantly LRU. The best performance is noticed when neither the target nor the remote cache distort the objects’ popularity ranking by updating their contents or SW (Protocol 2.A.II).

The performance of both LRU and SG-LRU is dominated by the global hit rate, especially as the Zipf exponent increases, due to the absence of popularity ranking filtering.

Surprisingly enough, LRU outperforms slightly SG-LRU in the (unrealistic) scenario where the cache size equals the catalogue size. This seems to be related with the time needed by the SW-LFU function to collect the most popular objects in the cache.

In Protocol 2.B, where the remote cache updates both its contents and its SW upon a global hit while the target cache does not update its contents, SG-LRU performs better when the cache is small and the “global statistics” are utilised by enabling the target cache to update its SW (Protocol 2.B.I) than when this operation is prohibited (Protocol 2.B.II).

Finally, when Protocol 2 is used, the utilisation of a large SW benefits the caching efficiency.

**General Remarks**

The most interesting conclusion of our study is perhaps that SG-LRU outperforms LRU under any Protocol, provided that the SW size has been selected appropriately.
Also, the following design guidelines are of major importance:

- When our goal is to achieve the maximum composite cache hit rate possible, then we can apply Protocol 2. The penalty we have to pay in this case is the higher cooperation overhead.
- When our goal is to minimise the cooperation burden, we can apply Protocol 1. This might lead to more often downloading of the requested objects from the origin servers.

Finally, we should note that when the caches have large storage capacity, then it is the local hit rate that dominates the performance of the system. On the other hand, when the nodes are equipped with small caches, it is the distributed cache as a whole that determines the system performance.

### 5.5 Cache-Enabled Opportunistic Joint Transmission

#### 5.5.1 Introduction and Motivation

Joint Transmission (JT) constitutes a radio communication technique where a set of cooperating BSs share the user data (possibly along with control information) over the MBH, in order to jointly serve the scheduled users [6]. As a result, the co-channel interference (CCI) is eliminated and a spatial multiplexing (SM) gain is provided. On the downside, JT places an extreme burden on the backhaul, due to the requirement for data exchange between the cooperating nodes.

On the other hand, the availability of user data at the caching nodes gives rise to JT opportunities while at the same time it addresses the data sharing issue. Nevertheless, content placement should promote the duplicating of content over different caching nodes in order to create such JT opportunities [15][16]; otherwise, the probability of finding the requested content in more than one caching nodes, so that JT could be applied, is negligible [16].

The Hybrid SG-LRU scheme described in Section 5.3 promotes such content replication, due to the use of global popularity statistics in caching decisions. Similarly, as we discussed in Section 5.4, some variants of the proposed cooperative reactive caching schemes enforce such content duplication (namely, the ones that follow the Protocol 1 approach), whereas others (i.e. the ones that follow the Protocol 2 approach) do not allow the replication of objects at different caching nodes. More specifically, all the variants of Protocol 1 enable the update of the target cache (as well as of its SW) upon a global hit.

Note that Hybrid SG-LRU does not involve any data exchange between the caching nodes. The level of “convergence” (i.e. content duplication) among the different caches is a consequence of local scores sharing instead (i.e. use of global statistics). Similarly, there exist variants of Protocol I in Collaborative SG-LRU where the caching nodes update only their SW. Moreover, in variants that
allow cooperative update of the cache contents, this take place only upon a local cache miss, in which case the requested object would be anyway downloaded from the origin server in the absence of some cooperation mechanism. Instead, in this setup is fetched by a nearby cache server, so that the delay and total network bandwidth consumption is reduced.

### 5.5.2 Numerical results

In this Section, we study via numerical simulations the ability of the several collaborative caching variants described in Section 5.4 as well as of the Hybrid SG-LRU strategy described in Section 5.3 to enable JT through exploitation of common data cached at different nodes. We assume a catalogue of size $N = 1,000$, a composite cache of physical size $M = 100$ objects (i.e. 10% of $N$) corresponding to $U = 5$ caches with storage of size $M_0 = M/U = 20$ objects (i.e. 2% of $N$), a Zipf shape factor $\alpha = 0.75$, $R = 100,000$ requests at each cache and $S = 10,000$ simulation runs. The performance metric of interest is the average ratio of JT opportunities after all simulation runs, i.e. the average number of times where the employed caching variant enables the utilisation of JT over the total number of requests, expressed as a percentage.

![Figure 5-32. Ratio of JT opportunities for Hybrid SG-LRU and different collaborative caching variants.](image)

[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017
We note that the different flavours of Protocol 1, which enables content duplication at different caching nodes in contrast to Protocol 2, achieve an average JT opportunities ratio of about 25\%, while the various flavours of Protocol 2 either do not create JT opportunities or they achieve a very small average JT opportunities ratio (< 4\%). Also, Hybrid SG-LRU gives rise to JT opportunities in the 19.84\% of the time.

5.5.3 Conclusions

In this Section, we studied the ability of Hybrid SG-LRU and Cooperative SG-LRU to create JT opportunities. The simulation results revealed that these caching schemes can give rise to JT transmissions about 20 - 25\% of the time, without involving the data sharing overhead of conventional JT techniques.

6 Summary and Conclusions

In this Deliverable, we studied the application of multicast beamforming as a means to enable the distribution of popular multimedia content towards the terrestrial distribution network in an efficient manner.

Initially, we proposed a framework that facilitates the application of satellite multi-group multicasting at the hybrid satellite-terrestrial network considered in SANSA.

Then, assuming a full frequency re-use multi-beam architecture, we studied a proposed multi-group multicasting aware frame-based precoding scheme that considers per-antenna transmission power constraints. We assumed partial cooperation (i.e. local CSI sharing) between a number of interconnected satellite gateways. Also, beam clustering is assumed. The simulation results showed that the proposed technique, which alleviates the capacity limitations of the feeder link, provides good availability to the backhaul nodes. Moreover, it approaches closely the energy efficiency of the single gateway upper bound in the low power regime, where the inter-cluster interference level (which is the main performance limitation in this setup) is low.

Next, we studied a proposed hybrid satellite-terrestrial proactive caching scheme that utilises both satellite multi-group multicast and terrestrial unicast transmissions in order to update the contents of the caches at off-peak hours. This caching strategy makes use of both local object popularity information obtained at each caching node as well as global object popularity information obtained as the average of the local popularities over all caches (mono-beam architecture) or over the caches located at the same spot beam / cluster (multi-beam architecture). The simulation results showed that the proposed hybrid satellite-terrestrial architecture approaches closely the caching efficiency of the terrestrial-only upper bound while at the same time it reduces significantly the time required...
for content placement, thanks to the exploitation of both local and object content popularity information and the utilisation of satellite multi-group multicasting transmissions, respectively. Furthermore, the simulations indicated that the multi-beam architecture outperforms the mono-beam one, due to the more accurate representation of the global popularities in the former case.

We continued with the study of a proposed reactive caching scheme, called Hybrid Score-Gated Least Recently Used (SG-LRU), which complements the aforementioned proactive caching strategy. This scheme can be applied to a multi-cache system where the caches are divided into disjoint clusters (e.g. the caches located at a spot beam may define a cluster). This caching policy utilises an LRU cache and a Sliding-Window Least Frequency Used (SW-LFU) score-gate function, which stores the reference count of each object over a recent timeframe (SW size). The score-gate serves as an admission control function, in order to avoid the frequent loading of objects into the cache and the storage of “one-timers”; as a caching efficiency booster, since it collects in the cache the most popular objects; and as an aging mechanism, since the object request frequencies are considered in a finite sliding backlog, in order to provide adaptation to content popularity shifts and eliminate the pollution of the cache with outdated objects. The LRU cache, on the other hand, allows for simple implementation and low, constant $O(1)$ update effort per request. The size of the sliding window determines the behaviour of this caching policy. For small values of this parameter, the performance of this caching scheme resembles that of LRU, while for large values it is similar to LFU. The latter is the optimum strategy, in terms of caching efficiency, under the Independent Reference Model (IRM) but due to its unlimited backlog of object request frequencies it cannot react to object popularity dynamics and it presents extreme computational and storage space requirements that prohibit its use in practical implementations. The Hybrid SG-LRU makes use of both local and global scores in order to calculate a total score for each object, with the local score being the request count for that object within the sliding window timeframe calculated at the caching node of interest and the global score being the average of the local scores calculated at the remaining caches of the same cluster. A weight determines the local score vs. global score balance in the computation of the total score of each object. The simulation results revealed that the Hybrid SG-LRU scheme outperforms significantly the LRU strategy, especially for small caches, small to moderate Zipf exponents, relatively large sliding window, and moderate to large weight, and approaches closely the LFU cache hit rate.

In the following Section, we studied a family of collaborative LRU / SG-LRU based caching schemes that differ on whether they allow or prohibit the target or / and remote cache to update its cache or /and its sliding window upon a global cache hit. The simulation results revealed that under appropriate selection of the sliding window size, the SG-LRU variants outperform significantly their LRU counterparts and approach closely the optimum LFU cache hit rate.
Finally, we demonstrated via numerical simulations the ability of the aforementioned Hybrid SG-LRU and Collaborative SG-LRU variants to enable opportunistic joint transmission while alleviating the need for user data exchange over the mobile backhaul, which is attributed to the replication of content across different caching nodes due to the utilisation of global statistics or specific cooperation directives that enforce such content replication.

The work presented in this Deliverable provides a pragmatic view and a holistic solution in the exploitation of satellite multicasting in order to enable the efficient distribution of popular multimedia content over hybrid satellite-terrestrial backhaul networks.
References


[D4.4: Multicast beamforming for distribution of popular multimedia content towards the terrestrial distribution network]

Date: 28/07/2017


Satellite Network of Experts (SatNEx) 3, “Call of Order 2 – Task 1: Fair Comparison and Combination of Advanced Interference Mitigation Techniques,” *ESA Contract 23089/10/NL/CPL*.


