NOTED: a framework to optimise network traffic via the analysis of data from File Transfer Services

Joanna Waczyńska[1,2,*], *Edoardo* Martelli[2,*], *Edward* Karavakis[2], and *Tony* Cass[2]

[1]Wrocław University of Science and Technology (Wroclaw, Poland) – email: `joanna.waczynska@gmail.com`
[2]CERN (Geneva, Switzerland), IT department – email: `firstname.lastname@cern.ch`

Abstract. Network traffic optimisation is difficult as the load is by nature dynamic and seemingly unpredictable. However, the increased usage of file transfer services may help the detection of future loads and the prediction of their expected duration. The NOTED project seeks to do exactly this and to dynamically adapt network topology to deliver improved bandwidth for users of such services. This article introduces, and explains the features of, the two main components of NOTED, the Transfer Broker and the Network Intelligence component. The Transfer Broker analyses all queued and on-going FTS transfers, producing a traffic report which can be used by network controllers. Based on this report and its knowledge of the network topology and routing, the Network Intelligence (NI) component makes decisions as to when a network reconfiguration could be beneficial. Any Software Defined Network controller can then apply these decision to the network, so optimising transfer execution time and reducing operating costs.

1 Introduction

The goal of the NOTED (Network Optimized Transfer of Experimental Data) project is to predict the arrival and duration of large data transfers and, if and where relevant, to suggest appropriate changes to optimise the network configuration and so reduce transfer duration. One possible network optimisation is to load-balance traffic between the main path and an alternative such as a backup path or an unused connection [1]. Alternatively, a larger capacity link provided by a dynamic provisioning service could be activated.

Unfortunately, detection of network congestion alone cannot lead to a proposal for an appropriate network optimisation. Such proposals require knowledge of at least the source and the destination of a transfer and also its predicted duration—there is no point proposing an optimisation for a transfer that will complete before the suggestion can be implemented. For this reason it is also desirable to be able to predict when a large transfer will start so that the network configuration can be prepared in advance.

The NOTED project aims to exploit information from FTS (File Transfer Service [2]), the main file transfer service used by WLCG, to gain the necessary knowledge and to transform this to a format that can be understood and exploited by network operators. To more precisely identify the source and destination of a data transfer, NOTED enriches the FTS transfer information with network addresses extracted from the CRIC (Computing Resource Information Catalog [3]) database used by the LHC experiments.

[*] main authors

2 NOTED architecture

There are three main components of NOTED.

- The **Transfer Broker (TB)** extracts transfer information from FTS and enriches these with network information from the CRIC database. Although FTS manages transfers independently, the TB is capable of aggregating FTS information to identify bulk data transfers between sites. The TB produces its output in json format.
- The **Network Intelligence (NI)** component interprets information provided by the TB to identify large transfers, and using its knowledge of the network topology and how transfers will be routed, proposes an appropriate network optimisation such as activating a backup link or adding an alternative path. The NI is also responsible for deciding when a network optimisation is no longer needed.

  The NI has multiple "controllers" which operate in parallel, each focusing on one or more specific network path(s). The NI controllers are provided with relevant source and destination pairs which are then used to select a subset of the TB information. It should be noted that the NI controller configuration needs to define not simply the end points of its network path(s) but all possible sources of, and destinations for, traffic that will traverse these paths. This is because network links can be shared by multiple sites and congestion generally occurs when multiple transfers converge on a single path segment. We will show later that the ability of the NI to aggregate information concerning multiple transfers that cross a particular network link greatly improves the prediction of network congestion.

  A complex Convolutional LSTM model [4] is used to analyse TB information to predict the overall load for a network segment. We estimate the expected duration of a high load situation using linear regression as explained later.
- **Software Defined Network Controllers (SDNC)** that are responsible for re-configuring the networks they control, guided by triggers from the NI, and so delivering better performance and reducing the duration of file transfers.

3 The Transfer Broker

In order to better understand the network flows, the first step is to understand what is generating the traffic. For WLCG, most network traffic is generated by FTS. We therefore analyse the data transfers generated by FTS in order to estimate when any network optimisation should be applied.

The first component, the Transfer Broker (TB), is responsible for collecting and analysing data from FTS, and for publishing grouped transfer information on a website that can be accessed by third party network controllers. Figure 1 presents the operation and main components of the TB. The TB information is made publicly available in a json file. The structure of this json file is described in figure 2. The role of FTS is to manage bulk data transfers on behalf of users. To do this, FTS maintains a list of files to be transferred and copies these from the source storage to the destination storage, limiting the number of simultaneous transfers to ensure network and storage system usage is maintained within set limits, FTS thus maintains a queue of files to be copied which gradually empties as file transfers complete.

The first implementation of NOTED estimated the impact of future data transfers by checking files in the FTS 'SUBMITTED' state [1]. Multiple tests have showed that this approach to the problem wasn't good enough to correctly detect large transfers. In our improved approach, we also take into account the FTS history of all transfers where the last report took place not less than a given amount of minutes ago. The history of transfers helps us to better understand the whole transfer operation (for example the throughput trend). As a result, our transfer classification is more accurate, because it is based on the whole t_h period, and not only the last report.

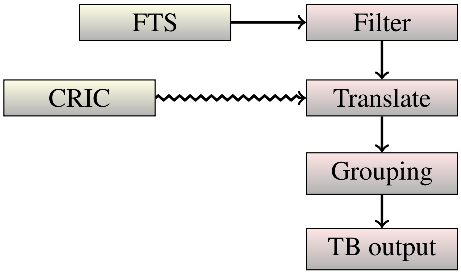


Figure 1: Components of the Transfer Broker.

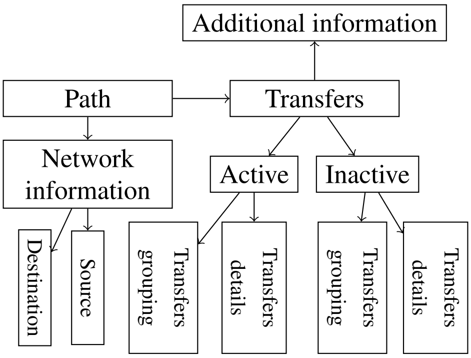
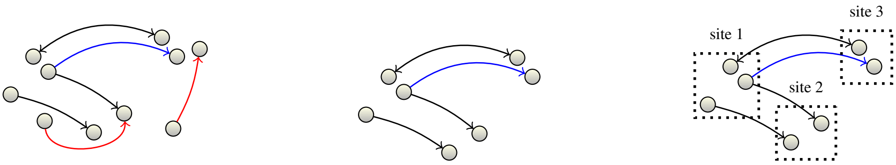


Figure 2: TB output scheme.

A further point to consider is that FTS has its own procedures for handling transfer failures and other errors and the TB needs to take into account the different ways in which these affect the information it receives. The FTS Optimiser is aware that FTS is not the only user of shared networks and storage servers and that pressure on resources used by FTS could be a cause of transfer failures. Such cases are recoverable and FTS will therefore reschedule the transfers hoping that the resource contention has gone away. However, there are other causes of failure that are unrecoverable and for which, although the requests remain queued, FTS will not re-attempt any data transfer. The TB uses adjustable heuristics (based on general guidelines in the FTS documentation) to filter transfers into two groups, active and inactive (figure 3a and 3b), where the active group should contain all transfers except those with an unrecoverable failure. It should be noted that transfers with a recoverable failure due to network saturation are of particular interest as they indicate that a network reconfiguration could be beneficial.

Finally, the TB uses information from the CRIC database [3] to identify the site network prefixes (ipv4/ipv6) of the storage elements involved and then groups transfers based on this information, as shown in figure 3c, since this is the level at which a network controller can take action.



(a) Transfers between endpoints. Red represents inactive transfers, black active, blue transfer which is now inactive, but it was active.

(b) Filter transfers: inactive transfers are discarded. This operation could be defined by input settings.

(c) Translate FTS endpoints to unique sites using CRIC, then transfers are grouped using {source,destination} key.

Figure 3: Identification and grouping of relevant transfers.

4 The Network Intelligence

The Network Intelligence (NI) component is a topology-aware system that interprets information from the Transfer Broker and can signal the start and stop of large aggregated transfers affecting a given link/path.

Its main components are presented in figure 4. Inputs to this pipeline are: a) the Transfer Broker output with all information about transfers and sites; and b) the controller configuration which sets up the control parameters used for the analysis, decision making and predictions for the different network segments of interest.

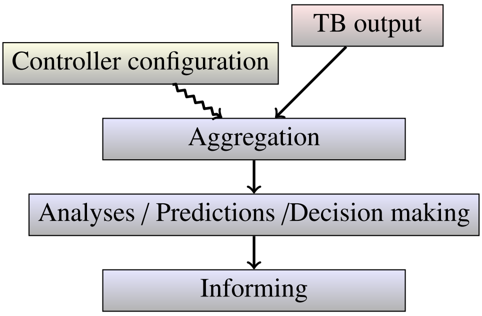


Figure 4: Components of the NI.

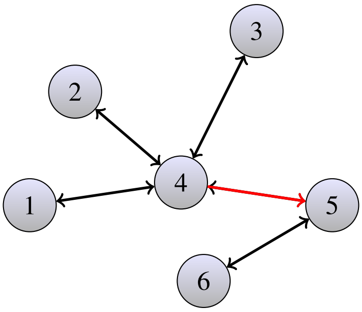


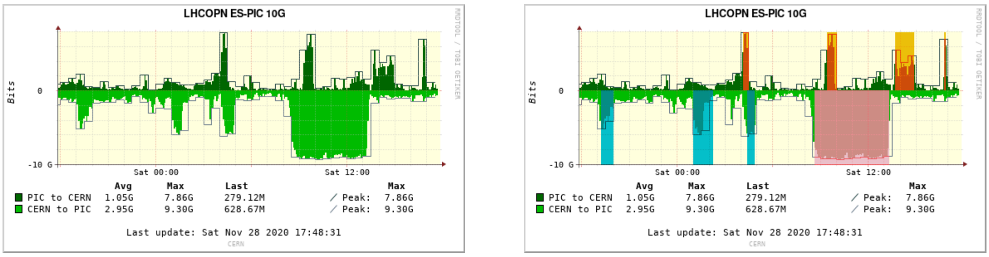
Figure 5: Example of network topology.

The aggregation is important in order to better understand the FTS decisions. The accumulated data gives us an insight into the current situation. We often are not able to see the impact on network traffic of a single large transfer, which the transfer broker may have qualified as important. In fact, visible increments of network traffic are caused by multiple transfers. Knowing the topology and the routing policies is important, because a network link to a given destination could be used not only by the directly connected source, but also by other remote sources with the right to transit. The NI first classifies these transfers into the same group.

The aggregation stage is a key element of the NI process as it enables us to combine the impact of potentially independent FTS decisions to give information about the impact on one or more network path segments. This is illustrated in figure 5. To understand the traffic flow along the red path ($\psi_{4,5}$), we have to aggregate data according to this equation: $\psi_{4,5} = \sum_{i=1}^{4} \sum_{j=5}^{6} \tau_{i,j}$, where $\tau_{i,j}$ are transfers from i to j. We can see that path $\tau_{4,5}$ is also influenced by paths $\psi_{i,4}$, where $i \in \{1, 2, 3\}$. Suppose all tracks have the same capacity. Sending a transfer from 1 to 5, we have to consider $\psi_{1,4}$ and $\psi_{4,5}$ paths. The maximum of transfer throughput will be the minimum of the available throughput ($\psi_{1,4}$ and $\psi_{4,5}$).

Due to the behaviour of the FTS Optimizer, there is, on an empty or lightly-loaded network path, a linear relationship between FTS transfers and network traffic. Although different network paths have distinct properties (such as capacity and RTT) and FTS is not the only source of traffic (even on WLCG networks) we assume that this linear relationship still holds.

The NI defines so-called start and stop moments. These are the points in time at which a network controller should apply and remove a network reconfiguration that can improve a transfer. The rules for creating the start and stop moments are not hard-coded, but defined by the configuration input. In this way, we can easily adapt the behaviour of the NI controllers based on experience.



(a) Utilization of CERN-PIC link on 27/28-11-2020. FTS was the main traffic source.

(b) Large transfers detected by the NI.

Figure 6: CERN-PIC 10Gbps LHCOPN link. Source: CERN network monitoring website [5]

The ability of our NI to identify times at which network optimisation could be of benefit is well illustrated in figure 6 which covers a period when the NI was monitoring production traffic. After aggregating transfers and combining information from the TB, the NI made several decisions. The red area indicates the time when the NI considered the LHCOPN ES-PIC link to be saturated and that load balancing over additional links could have had a positive effect. The orange area and blue areas highlight times when the NI identified load from, respectively, small- and medium-sized transfers but did not recommend network reconfiguration. Comparison with the actual link utilisation demonstrates that the NI can fully understand network loads when the bulk of the traffic is generated by FTS.

5 Software Defined Network Controllers

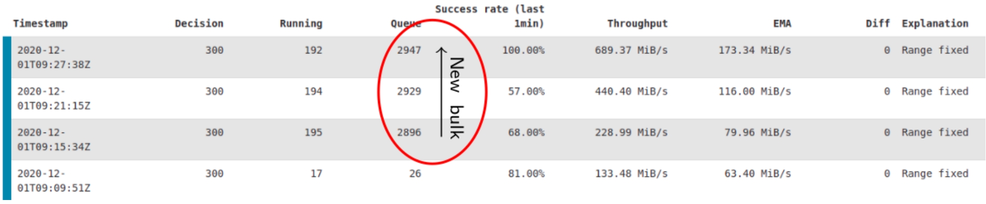
The Software Defined Network Controller (SDNC) [6] is the component that actually modifies the configuration of network devices, implementing and removing special routing policies at the time suggested by the NI to https://www.overleaf.com/project/603908c888d36781d7fd1002improve the performance of data transfers. For example, a data transfer could be load-balanced over the primary, usual link and a second link which may have a different purpose, but it is under utilized at that moment. Or, the NI could request a temporary network circuit of a higher capacity to be provisioned between the source and the destination of the transfer.

It should be noted that there is an interesting feedback loop between the SDNC and the NI. The goal of the changes proposed by the NI is to reduce the duration of a large transfers. A successful network reconfiguration will lead to improved transfer bandwidth and thus invalidate the NI's initial estimate of the time of completion. The NI must therefore regularly re-calculate the time of completion to ensure that the network optimisation is removed as soon as it is no longer necessary, and not simply left in place until the original estimation of the unoptimised transfer completion time.

6 Applications

To date, NOTED has been tested by checking FTS transfers to WLCG Tier1s. This is because all their LHCOPN links are connected to CERN and so can easily be controlled by a CERN-based SDNC. Multiple tests have been performed over the past two years but we concentrate here on one carried out on the 1st December 2020 between CERN and PIC, the Spanish WLCG Tier1.

A controlled transfer generated by the ATLAS experiment started at 9:15 (UTC). The transfer was injected using Rucio [2, 7], hence it was presented by the FTS monitoring website (figure 7). The column `Running` reports the number of active files; the column `Queue` reports the number of submitted files. At the beginning of the ATLAS transfer, FTS was moving 31 other files (around 50Gb in total) across the same link for another transfer. That means that a previous transfer hadn't finished when our CERN-PIC ATLAS transfer started.



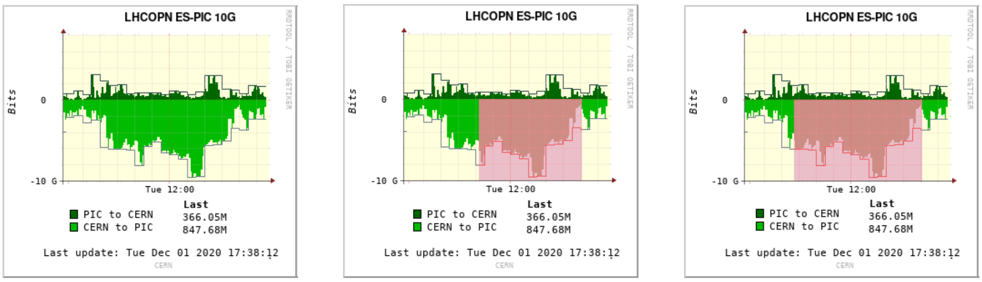
Timestamp	Decision	Running	Queue	Success rate (last 1min)	Throughput	EMA	Diff	Explanation
2020-12-01T09:27:38Z	300	192	2947	100.00%	689.37 MiB/s	173.34 MiB/s	0	Range fixed
2020-12-01T09:21:15Z	300	194	2929	57.00%	440.40 MiB/s	116.00 MiB/s	0	Range fixed
2020-12-01T09:15:34Z	300	195	2896	68.00%	228.99 MiB/s	79.96 MiB/s	0	Range fixed
2020-12-01T09:09:51Z	300	17	26	81.00%	133.48 MiB/s	63.40 MiB/s	0	Range fixed

Figure 7: Report between two endpoints. New bulk (our test transfer) started at 9:15 (UTC). Test transfer contained 3060 files (avg file size >6Gb); totalling around 145 Tb.

Two different NI controllers were compared, in order to check the influence of different data aggregation calculations on decisions made by the NI during the test. Both controllers tagged transfers using the same rules. The first, however, focused on just the transfers between CERN and PIC, read directly from the FTS; with the second also collecting information from all the declared sites from which transfers to PIC could pass through CERN. Figure 8 presents the operation of the two NI controllers. The red area that highlights the increased traffic caused by FTS is wider on figure 8c than on figure 8b. This is because the traffic between 9:30 to 10:00, which the first NI controller did not pick up, was not generated at CERN, but was rather transiting CERN to reach PIC. This shows that data aggregation, i.e. aggregation of all transfers which, according to the network topology, can pass through the observed source (in this example: CERN) to a given destination (in this example: PIC), is necessary for an accurate characterisation of the network traffic.

The action taken by the SDNC to increase the available bandwidth between CERN and PIC was to add the LHCONE path in parallel to the CERN-PIC LHCOPN link and to load balance the traffic over the two links. The CERN-PIC traffic was thus split into two paths (half on LHCOPN_ES_PIC, and half on LHCONE_GEANT). The impact of manually removing the additional path (LHCONE_GEANT) is presented in figure 9. As expected, without the extra path, the LHCOPN_ES_PIC path became overloaded.

Ideally, traffic should have reached 20Gbps to properly demonstrate the usefulness of the network reconfiguration but, for reasons beyond our control, this was not possible. Hence, the rate of the FTS transfers was constant and we were able to predict the transfer finish time using linear regression. An ordinary Least Squares linear model was a good fit for this data set and predicting the exact finish time was possible from the point where 40% of the transfer had been accomplished (figure 11). To make the prediction, the NI analysed the estimated value of the filesize of the queue and the active files reported by the TB (figure 10b). However, we must remember that the use of a linear regression model is only possible under specific conditions. First of all, we expect that as soon as we add an additional link, we will be able to increase the bandwidth, increasing the transfer speed. as a result, a linear relationship will be observable only by segments. Therefore, linear regression model will not be a good fit for the entire data set [7]. Nevertheless, linear regression is a good baseline model [8].

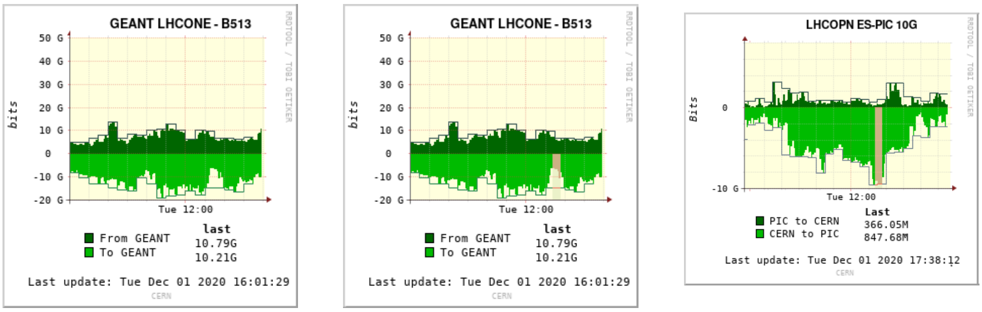


(a) Observation of traffic passing through the 10Gbps path from 01-12-2020. Multiple overlapping transfers can be observed.

(b) This NI controller focused on transfers from CERN. It detected our test transfer and recommended network optimiisation only for that.

(c) The second NI controller detected and aggregated, in accordance with figure 5, all FTS transfers transiting CERN to reach PIC, including those generated by other Tier1s. Network optimisation was thus recommended for a longer period.

Figure 8: Operation of two NI controllers. The red colouring shows increased traffic due to transfers from FTS.

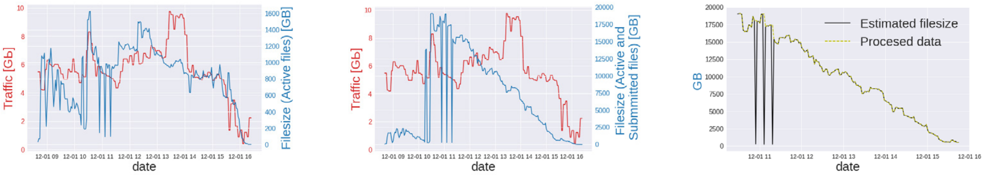


(a) Observation of traffic passing through the added LHCONE_GEANT path from 01-12-2020.

(b) The yellow area presents the period when LHCONE_GEANT path wasn't used to load-balance the traffic between CERN and PIC. The Network traffic decrease in the LHCONE_GEANT path is visible.

(c) The yellow area presents the period when the LHCONE_GEANT path wasn't used to split traffic between CERN and PIC. Network traffic increased in the LHCONE_ES_PIC path.

Figure 9: Impact of removing the added path on the network traffic of the observed links.



(a) Estimated filesize of active files.

(b) Filesize of active and submitted files from high-priority transfers.

(c) Processing data during the test transfer.

Figure 10: Estimated filesize of queued and active files based on aggregation of all transfers from FTS from selected sites. Data set from second controller.

Analyses and decisions during the test were based on transfers that TB tagged as active. Active transfers were defined by rules: Average of the last quarter `success_rate` was higher than 74%, or the last `success_rate` was higher than 74%. The last `fts_decision` was higher than 2. Information about transfer had to be updated in the last 10 minutes. On the FTS report (figure 7), we can notice that at the beginning of the test, `success_rate` went down to 57%, which means FTS had a problem with sending packages correctly. TB classified our test transfer as "inactive", so in theory we could have lost information about this transfer. However, this did not happen because the NI has the ability to postpone making path removal decisions during the transfer. In our case, this additional time was set to 10 minutes. The waiting time for the next report was 6 minutes. Also the three drops in figure 10c show that we lost information about the transfer several times during transfers. Thanks to the additional functionality, we avoided unnecessary reconfiguration.

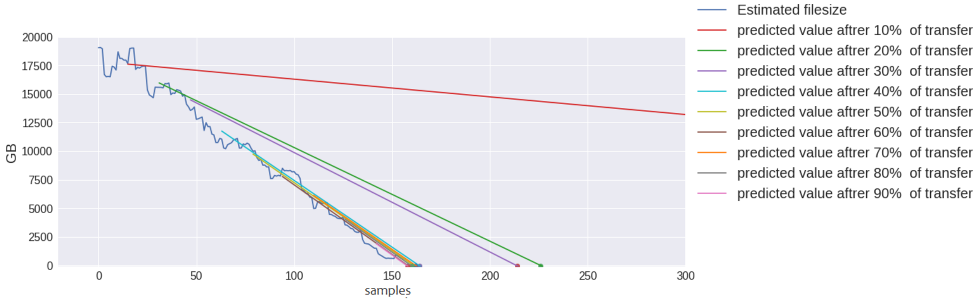


Figure 11: Predicted values based on specific parts of data being accomplished.

Although after aggregation, the test transfer had the highest influence on the data set, the occurrence of additional transfers showed clearly that aggregation is an important step in analyzing network traffic (figure 8). Tagging transfers with specific rules also had a significant role. The effects can be seen in figure 10. Additionally, we have to remember that on that day we also observed network traffic not generated by FTS. All these effects influenced the accuracy of the test. However, the tool, despite the difficult data set, responded correctly.

7 Conclusion

The NOTED project has demonstrated that exploiting data from transfer services such as FTS can indeed enable the successful prediction of network loads. This was clearly shown in figure 6. We have also demonstrated that we can correctly aggregate information concerning transfers from multiple sources to predict load on a particular network link (figure 8) and further that such aggregation is necessary to correctly characterise network loads. Finally, we have demonstrated that we can exploit our understanding of network traffic loads to reconfigure the network to deliver improved utilisation and faster data transfers.

The cognitive analysis of the behavior of a specific network could allow a more accurate understanding of the network traffic caused by the FTS transfers. Therefore, the next step should be the development of nonlinear models [4].

References

[1] C. Busse-Grawitz, E. Martelli, M. Lassnig, A. Manzi, O. Keeble, T. Cass, *The NOTED software tool-set improves efficient network utilization for Rucio data transfers via FTS.*, EPJ Web of Conferences **245** (2020)

[2] E. Karavakis, A. Manzi, M.A. Rios, O. Keeble, C.G. Cabot, M. Simon, M. Patrascoiu, A. Angelogiannopoulos, *FTS improvements for LHC Run-3 and beyond*, EPJ Web of Conferences **245** (2020)

[3] C.D.T. at CERN, *Cric computing resource information catalog. core service/site*, http://wlcg-cric.cern.ch/ (2020), accessed: 17.12.2020

[4] J. Waczyńska, E. Martelli, S. Vallecorsa, E. Karavakis, T. Cass, *Convolutional LSTM models to estimate network traffic*, Paper presented to vCHEP 2021s (2021)

[5] *Lhcopn and lhcone traffic*, https://netstat.cern.ch/monitoring/network-statistics/ (2020), accessed: 18.12.2020

[6] S.M. Marian Babik, *Network Capabilities for the HL-LHC Era*, EPJ Web of Conferences **245** (2020)

[7] J.I. Bogado Garcia, M. Lassnig, I. Vukotic, F. Monticelli, *Modelling High-Energy Physics Data Transfers-slides*, Tech. rep., ATL-COM-SOFT-2018-152 (2018)

[8] S. Date, *Fitting linear regression models on counts based data*, towardsdatascience.com, accessed: 06.01.2021