

E-Stream: Towards Pattern Centric Network Incident Discovery and Corrective Action Recommendation in Telecommunication Networks

Sebastian Robitzsch*, Faisal Zaman*, Zhiguo Qu*, John Keeney[†], Sven van der Meer[†] and Gabriel-Miro Muntean*

*The Rince Institute

Dublin City University, Dublin, Ireland

Email: (sebastian.robitzsch, faisal.zaman, zhiguo.qu, gabriel.muntean)@dcu.ie

[†]Ericsson

Network Management Lab

Athlone, Ireland

Email: (john.keeney, sven.van.der.meer@ericsson.com)@ericsson.com

Abstract—With the technological evolution in telecommunication networks, performance requirements such as better coverage, higher bandwidth, and lower latency have been pushed to new horizons. However, as a direct result network complexity has increased dramatically over the recent years, and with this complexity manageability has suffered. This paper presents the architecture of the E-Stream project which aims to support Next Generation Operations Support Systems. E-Stream applies dimension reduction, data mining, and recommender system techniques in order to handle very high volumes of management events, identify and predict network incidents, and recommend candidate corrective actions to domain experts in Network Operations Centres.

I. INTRODUCTION

Over the recent years the amount of data traffic generated by users accessing the Internet via carrier-grade Radio Access Technologies (RATs), e.g., GSM, UMTS or LTE, has increased exponentially [1]. Not only does every individual customer consume more bandwidth, but also the number of active mobile subscribers has increased world-wide reaching 6.2 billion mobile broadband subscribers [1]. This significant increase in demand is addressed by more advanced network architectures like LTE and LTE-advanced, which aims to deliver services according to the subscriber requirements. However, with an increase in more advanced functionalities, which must still co-exist with current network technologies, e.g., 2G, 3G or Wi-Fi, complexity is also increasing significantly [2]. Consequently, the required maintenance of such networks becomes more and more a challenge for the operators, both technically and financially.

Current Operations Support Systems (OSSs) are capable of detecting pre-defined network incidents, e.g., cell congestions, by monitoring certain thresholds in various Network Elements (NEs). Eventually, warnings and alarms are triggered to domain experts in the Network Operations Centre (NOC) in order to indicate an incident in the network. The domain experts then manually investigate the reports starting with the most critical ones. The process of resolving even non-trivial network incidents requires significant comprehensive knowledge about the network architecture, its elements and

their capabilities. Additionally, when it comes to more complex scenarios in which the root-cause relationship of a network incident is spread over gigabytes of traces, in different formats, from different sources, the time to resolve the incident by the domain experts increases disproportionately. As this type of scenario is not an exception, only network incidents of very high importance can be resolved. Therefore, as networks scale and become more complicated and current manual approaches prevail, it will be necessary to increase staffing for network management and operation; however, this is not feasible and such increased operational cost will not be tolerated. It is clear that the OSSs that monitor and manage these networks need to be increasingly automated [3].

In order to address this challenge, this paper proposes a framework which is capable of investigating trace information of a mobile network in a streaming fashion following an unsupervised data mining approach in order to discover any network abnormality. As studied in [4], there is a significant difference between offline processing of trace data with persistent relations, and online processing of streamed data which is continuous but varies in time and space. Thus, instead of performing a bounded one-time batch process, as it is implemented in current OSSs, a framework taking stream data as an input must work in a continuous and unbounded fashion. Such a system is much more time-critical due to a constant stream of incoming data which is continuous and not bounded. Real-time stream processing capabilities¹ are considered in this paper.

The remainder of this paper is structured as follows: section II describes the information flow within the E-Stream system and how it provides valuable information to the domain experts to significantly improve their time to resolve network incidents. Section III then presents the overall architecture of the E-Stream system including its modules and a description of their responsibilities; implementation challenges are also discussed. The paper is concluded in Section IV.

¹The term real-time describes the guaranteed completion of a process under a pre-defined time. The term does not imply the completion of a process with zero latency.

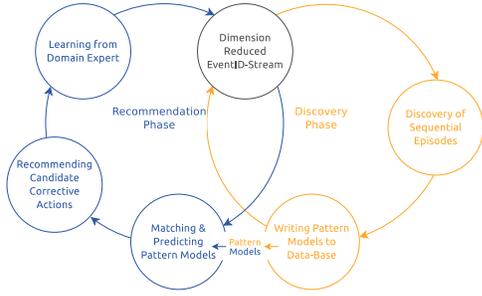


Fig. 1. Information Flow in the E-Stream Framework.

II. SYSTEM WORK-FLOW AND INFORMATION PROCESSING CYCLES

The next generation of OSSs require a more comprehensive multi-domain trace data analysis approach to allow operators a) to process and resolve network incidents more efficiently and in a more timely manner and b) to have the ability to predict future network incidents based upon prior network incidents and trends, and then recommend Candidate Corrective Actions (CCAs) indicating how the network incidents could be possibly resolved or mitigated.

The approach to perform such a task is to autonomously analyse trace files in the OSS regarding events that precede network incidents, which can be then used to try to evaluate a root-cause for the incident. Afterwards, a ranked list of CCAs should be presented to a domain expert in the NOC indicating possible solutions to a proactively detected incident. E-Stream utilises pattern discovery data-mining techniques to identify trends in the network trace stream, i.e., investigating sequences of events that preceded network incidents. Eventually, E-Stream should be capable of recommending CCAs to domain experts in the NOC by providing a list of suggested CCA, while continuously learning the most suitable CCAs for given patterns based on expert selections.

The process of finding patterns, providing a ranked list of CCAs and collecting the feedback from the domain expert can be divided into a *discovery phase* and a *recommendation phase*, as depicted in Figure 1. While both phases run completely independent from each other, they still share the same input data, i.e., a continuously incoming stream of trace data. However, both phases provide results in orthogonal time domains. While the recommendation phase acts on the same time-scale as the incoming data stream, the discovery phase requires some more in depth analysis of the data in order to find new patterns which can be provided to the recommendation phase. More real-time discovery can be applied cognisant of the need to trade-off resource usage and accuracy against timeliness. Scalability concerns can be addressed by the availability of High Performance Computing (HPC) infrastructures in OSSs.

III. E-STREAM SYSTEM ARCHITECTURE

The system architecture in Figure 2 depicts the E-Stream framework required to support the processing of a continuous stream of trace data from a mobile network. As the E-Stream work-flow is pattern discovery and recommendation centric, the identified architectural modules for realising this approach are as follows:

- Dimension Reduction Module
- Episode Discovery Module
- Episode Classification Module
- Pattern Matching Module and
- Recommender System Module

A. Architecture Description

In Figure 2 all defined E-Stream modules are drawn in orange. The data plane and control plane are coloured with solid blue lines and black dotted lines, respectively. Both data and control planes using arrows to indicate the direction of the information flow.

While streamed trace data from thousands or millions of NEs arrive at the framework, the Dimension Reduction Module (DRM) needs to create a one-dimensional representation of the trace stream, referred to as data stream. The data stream consists solely of numeric values due to the requirement of off-the-shelf pattern discovery algorithm working with number values only. In addition to discovering and merging the important fields in each event to generate a numeric value for each distinct event type, the DRM module is also responsible for detecting and removing events that can be considered purely periodic or noise, and so contribute little to the information content in the stream.

The single dimensional data stream is then analysed by the Episode Discovery Module (EDM) to find frequent closed sequences of events (episodes) E , e.g., $\{e_1, e_2, e_3\}$. Note, at the time when E was discovered, its meaning and mapping to further actions in later modules is still unknown, as E is just a sequence of numbers. Hence, E-Stream denotes those sequences as episodes instead of patterns. An episode is the frequent closed sequential set of events which need to be analysed further in order to become a pattern. A pattern can be divided into a predictive head and predicted tail, which can be defined as a sequence of one or more antecedents (head) followed by a consequent (tail) perhaps caused by the antecedents. The time to analyse a chunk of events from the data stream is denoted as t_{ed} .

To convert an episode to a pattern, E-Stream defines an Episode Classification Module (ECM) which classifies episodes and forms clusters of patterns into a relational pattern tree. Additionally, the ECM maintains the Pattern Model Library (PML) which holds all patterns. As depicted in Figure 2, the ECM has various control plane communication interfaces to other modules in order to receive module-specific pattern information which is used to classify the episodes and to provide related information to fine-tune other module's internal algorithms.

Once the PML is successfully populated with patterns, the Pattern Matching Module (PMM) searches the incoming data stream for known patterns. The key feature of the PMM is its ability to predict a full pattern based on a matched subset (head) of the predicted pattern, which then allows the E-Stream system to proactively indicate the likelihood of incidents prior to their occurrence, and preemptively recommend avoiding remedies. For the PMM to find match patterns in the data stream the pattern itself must have been discovered beforehand by the EDM and classified by the ECM. Since pattern matching can be efficiently implemented in Complex Event Processing

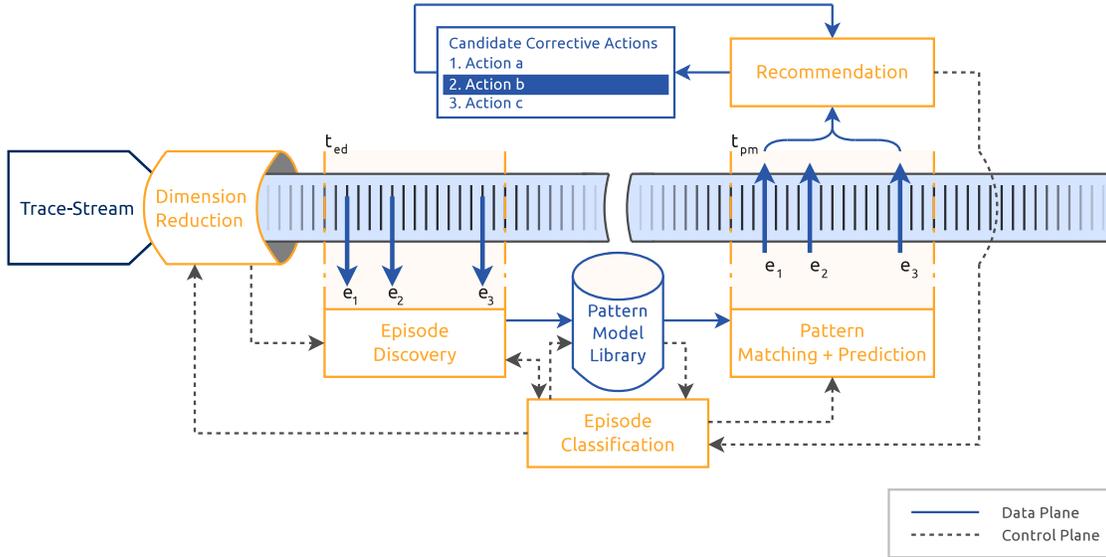


Fig. 2. Overall E-Stream Architecture Including Conceptual Flow of Trace Data and Control Messages.

(CEP) and Stream Processing (SP) systems, the time required by PMM to match and predict patterns in the stream, t_{pm} , is always significantly smaller than the time EDM requires: $t_{ed} \gg t_{pm}$. This is why the information cycle shown in Figure 1 depicts two completely distinct and independent processes.

Finally, the Recommender System Module (RSM) receives the matched and predicted patterns from PMM and recommends CCAs to the domain experts in the NOC. Those recommendations are learnt over time from previous recommendations. Not only does RSM learn possible CCAs for incoming patterns, it also reports back patterns that were used to resolve a problem which essentially helps ECM to further classify discovered episodes and to provide feedback about this to DRM, EDM and PMM.

B. Detailed Module Description

This section focuses on a more detailed description of approaches and algorithms within each module in order to achieve a module's task described in Section III-A.

1) *Dimension Reduction Module (DRM)*: DRM provides the necessary mechanism to reduce the dimensionality of the incoming trace stream from NEs in real-time by dynamic windowing, minimal-loss information reduction and online filtering. In order to achieve this goal, DRM uses the Pearson correlation coefficient to detect noise events in the stream in order to remove them so that the dimensionality of the stream is reduced, as published in [5]. In order to remove as little useful information as possible from the trace stream but decreasing the amount of data being produced, DRM receives feedback from ECM to learn about the importance of individual events and the context in which they are used. Therefore, DRM is capable of distinguishing between noise and important information to decrease the overall density in the data stream. Due to the requirement of real-time event processing, HPC environments such as Storm or S4 are most suitable to implement DRM.

2) *Episode Discovery Module (EDM)*: While offline data mining is usually performed over static input data of known length, EDM requires more advanced techniques to investigate the correlation between multiple events in order to report episodes [6]. However, many off-the-shelf data mining algorithms [7] [8] [9] to discover only frequent or infrequent patterns, for very high volume event streams, most frequent episodes, and their variations, denote normal behaviour, while most very infrequent episodes can be dismissed as noise episodes, derivative of frequent episodes, or artefacts of sampling. The EDM performs historical-oriented mining over multiple sampled windows in order to allow to detect interesting episodes. Additionally, E-Stream follows the approach of learning deviations from normal behaviour, where normal behaviour in a telecommunication network usually follows a network protocol pattern and interesting patterns are mostly derivatives from frequent patterns.

3) *Episode Classification Module (ECM)*: The classification of episodes into patterns is performed by the ECM and follows existing pattern classification techniques [10]. In addition to maintaining the PML, the ECM receives feedback information from the RSM about the popularity of patterns in order to train the DRM, the EDM and the PMM about the context the patterns have been discovered, matched and used. While the DRM is using this information to perform a more accurate spectral filtering, the episodes reported by the EDM can be further evaluated regarding their practicability. The PMM uses this information to fine tune its prediction mechanism. For instance, if a subset of a full pattern has been detected which is also a subset of another full pattern, the popularity ranking of predicted pattern is an important metric to decide on a more suitable candidate. This selection information can be gathered by the RSM to learn which patterns resulted in CCAs.

4) *Pattern Matching Module (PMM)*: In the recommendation phase the PMM acts as the pattern search engine which operates on the data stream. The main responsibility of PMM is to match patterns stored in the PML with the exact same

sequences of events in the trace. Second, it predicts future events using a pattern similarity criterion, i.e., predict the occurrence of the events which usually follow the matched sequence of events. The task of predicting an event based upon a known subset can be achieved by a tree-representation of inter-pattern correlations, [11], or similarity ranking [12]. The inter-pattern correlation calculation is accomplished by the ECM.

5) *Recommender System Module (RSM)*: The application of recommender systems is usually within the field of e-commerce or social media, e.g., Amazon, Facebook, Twitter or Spotify, where items get recommended to users based on various recommender system metrics. However, for recommending CCAs to domain experts with a high level of accuracy, RSM requires to identify how existing approaches can be mapped to E-Stream [13]. Instead of having an item-to-item, item-to-user or user-to-user relationship the RSM instead works on a pattern-to-pattern, pattern-to-domain expert and domain expert-to-domain expert relationship in order to achieve this goal. The RSM requires a hybrid recommendation approach (content-based and collaborative filtering) to accommodate all three categories into a single module [14] [15]. For instance, the RSM creates pattern and domain expert profiles to learn over time what domain expert has expertise in what area based on pattern categories he/she has solved before; calculating Term Frequency-Inverse Document Frequencies (TF-IDFs) for finding similar patterns is also an effective way to increase the reliability of the system. However, as outlined in various articles, collaborative approaches have shown significant lower performance benchmarks in terms of system response time. This can be expressed by Singular Value Decomposition (SVD) and matrix factorisation [16].

IV. CONCLUSION

In this paper presented E-Stream, a system to support next generation OSS architectures. E-Stream's goal is to find network trends in the trace streams from telecommunication networks to identify and predict network incidents. The incident report, predictions and trends are provided to domain experts in NOCs together with CCAs to prevent, fix or mitigate the incidents. As trace streams can vary significantly in space and time, the presented architecture was designed in such a way so that HPC frameworks can be utilised when implementing the concept. E-Stream defines five modules DRM, EDM, ECM, PMM and RSM which were described in further detail regarding potential algorithms and concepts that can be applied in order to meet each module's functional expectations. Although not discussed in this paper, the E-Stream approach, and its sub-modules are being tested in both test-bed [17] and emulated [18] environments.

ACKNOWLEDGEMENTS

This work was funded by Enterprise Ireland Innovation Partnership Programme with Ericsson under grant agreement IP/2011/0135 [19].

REFERENCES

- [1] Ericsson, "Ericsson Mobility Report - On the Pulse of the Networked Society," Tech. Rep., 2014. [Online]. Available: <http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf>
- [2] Ericsson, "Changing the Game before the Game Changes You," Tech. Rep., 2012. [Online]. Available: http://www.ericsson.com/res/docs/2012/discussion_paper_changing_the_game_e.pdf
- [3] F. Zaman, G. Hogan, S. van der Meer, J. Keeney, S. Robitzsch, and G. Muntean, "A Recommender System Architecture for Predictive Telecom Network Management," *IEEE Communications Magazine*, no. Status: Accepted for Publication, 2014.
- [4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and Issues in Data Stream Systems," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '02*. New York, New York, USA: ACM Press, Jun. 2002, pp. 1–16.
- [5] F. Zaman, S. Robitzsch, Z. Wu, J. Keeney, S. van der Meer, and G.-M. Muntean, "A Heuristic Correlation Algorithm for Data Reduction through Noise Detection in Stream-Based Communication Management Systems," in *IEEE/IFIP Network Operations and Management Symposium, (NOMS)*, 2014.
- [6] A. Bifet, *Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams (Frontiers in Artificial Intelligence and Applications)*. IOS Press, 2010.
- [7] X. Yan, J. Han, and R. Afshar, "CloSpan: Mining Closed Sequential Patterns in Large Datasets," in *SIAM International Conference on Data Mining (SDM'03)*, 2003, pp. 166–177.
- [8] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," in *7th International Conference on Data Engineering*, 2001, pp. 215–224.
- [9] T. Wang, M. Srivatsa, D. Agrawal, and L. Liu, "Spatio-Temporal Patterns in Network Events," in *Proceedings of the 6th International Conference (Co-NEXT'10)*, ser. Co-NEXT '10, vol. 5. ACM, 2010, p. 3. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1921172>
- [10] R. O. Duda, *Pattern Classification*. Wiley-Blackwell, 2004.
- [11] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs: Second Edition*, ser. Annals of Discrete Mathematics. Elsevier Science, 2004.
- [12] G. Salton and M. McGill, *Introduction to Modern Information Retrieval (Computer Science)*. McGraw-Hill Inc., US, 1983.
- [13] J. Keeney, S. van der Meer, and G. Hogan, "A Recommender-System for Telecommunications Network Management Actions," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2013, pp. 760–763.
- [14] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
- [15] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
- [16] Y. Song, L. Zhang, and C. L. Giles, "Automatic Tag Recommendation Algorithms for Social Recommender Systems," *ACM Transactions on the Web*, vol. 5, no. 1, pp. 1–31, Feb. 2011.
- [17] Z. Yuan, J. Keeney, S. van der Meer, G. Hogan, and G.-M. Muntean, "Context-aware heterogeneous network performance analysis: Test-bed development," in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*. IEEE, Mar. 2014, pp. 472–477. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6815252>
- [18] S. Robitzsch, "OpenMSC - An Open Source MSCgen-based Control Plane Trace Emulator for Communication Networks," 2014. [Online]. Available: <http://openmsc.blogspot.com>
- [19] Dublin City University and Ericsson, "E-Stream Project," 2014. [Online]. Available: www.estream-project.com