Anomaly Filtering in Large Scale Systems

Romaric Ludinard
Inria Rennes - Bretagne Atlantique (France)
Email: romaric.ludinard@inria.fr

***Abstract*—We consider a large number of users consuming services through Internet. When a problem occurs within the network, a large number of users perceive a quality degradation while a few number of them perceive it if the cause lies on user gateway or in the last mile. This paper describes how users can locally determine the impact of the problem. This is achieved by organizing users within a logical overlay whose design allows them to determine the impact of the problem they experienced.**

I. Introduction

Internet Service Providers (ISPs) bring access to the Internet to each of theirs clients. Each client expects a high quality and high reliability of its network access. Because of this natural expectation, an ISP has to monitor its network so as to be aware of its state and to be able to react when a problem occurs. However, due to the huge number of equipments to operate and the lack of efficient tools to track and detect anomalous behaviours, ISP relies on customer care call centres to be notified when problems are faced. This approach presents three limitations. It is expensive since ISP has to pay people to manually handle each customer notification, it may introduce a large latency detection since there is a potentially long delay between a problem occurrence and the user notification, and finally it may be inefficient since user may call its ISP for reasons that are not due to anomalous behaviours (plugging).

These limitations call for an automated procedure in order to notify ISP when a problem occurs. Pushing this monitoring task at the edge of the network allows to be as close as possible of the end user perception while leveraging end user activity. This approach was successfully adopted in [CBG10] but only focuses on massive problems. Actually, standardized procedures [Bro] exist at devices level to autonomously trigger investigations in presence of anomalies. However, these procedures are never used for practical reasons. Indeed if the cause of the anomaly lies in the network itself (e.g., at routers, links or data centre outages) this may impact a very large number of devices. For instance, if a VOD server becomes overloaded, each device consuming a video on this server will experiment a QoS degradation. In this case thousands of impacted devices will report the problem to the helpdesk operator and it may quickly become a disaster due to the volume of generated messages. In the same way, if a problem occurs in the last mile, only the end device concerned will suffer from a quality degradation and thus only one device will report the problem. It is thus of utmost importance to minimize the overall network footprint by giving each device the capability to self distinguish network-based anomalies from local ones – anomalies that only impact the device itself – so that only isolated anomalies are reported on the fly to the helpdesk.

The key point here is to provide each monitored device a way to estimate the impact of a perceived outage. In other words, each monitored device needs to find autonomously other devices that perceived the same problem and then decides locally whether it has to raise an alarm or not. The remaining of the paper is organized as follows. Section II briefly presents FixMe [ALS$^+$12] a self-managing and scalable architecture which allows monitored nodes to find nodes that experience similar quality variations. Section III describes the Anomaly Characterization Problem. Finally, Section IV presents ongoing investigations and futures works.

II. Finding similar nodes

Distributed Hash Tables (DHT) are classical approaches to deal with a huge population of nodes. In DHTs, each node is assigned a unique random m-bit identifier from a m-bit identifier space, derived from a standard one-way hash function (*e.g.* SHA-1 in [SMK$^+$01]). Nodes are organized into a structured graph according to their identifier. Hash functions uniformly distribute nodes over the identifier space and thus provide good properties to the structured graph while loosing the ability to compare nodes with their identifiers. Conversely, we proposed a novel architecture called FixMe where nodes organize themselves according to a general metrics which allows to find and compare nodes together while remaining scalable. In this way, nodes only need to know a very small part of the network to locally estimate the impact of a perceived outage.

In FixMe, we consider a set $\mathcal{N}$ of monitored devices that communicate to each other through the standard message passing model. Each node is assigned a unique random identifier derived from a standard one-way hash function (*e.g.* SHA-1). At any (discrete) time t, each device locally evaluates d quality metrics with an end-to-end performance measurement function $Q_i, 1 \leq i \leq d$. The quality range of each of the d measures is equal to $[0, 1]$. The quality perceived by a device j at time t is the vector $Q(j,t) = (Q_1(j,t), \ldots, Q_d(j,t))$. In addition to the functions $Q_1, \ldots, Q_d$, each node has access to d anomaly detection functions $A_1, \ldots, A_d$. At each time t, each function A_i is fed with the sequence of the $\ell_i \geq 1$ last quality values $Q_i(j, t - \ell_i + 1), \ldots, Q_i(j,t)$. Note that ℓ_i is a parameter of A_i. In this paper, we suppose that the output of these anomaly detections are boolean. At time t, $A_i(j,t) = \texttt{true}$ if the sequence $Q_i(j, t - \ell_i + 1), \ldots, Q_i(j,t)$ is considered as an anomaly, it is $\texttt{false}$ otherwise. Implementation of both Q_i and A_i functions are out of the scope of the paper.

All the monitored devices are mapped into a quality space $\mathcal{E} = [0,1]^d$, with $d \in \mathbb{N}$. Within $\mathcal{E}$, the position of a monitored device j at time t is the vector of the quality it perceived, and thus close points at time t are representative of devices exhibiting similar QoS at time t. For the quality metric $i = 1, \ldots, d$, we split interval $[0, 1]$ into n_i disjoint elementary intervals $[x_i^{(j-1)}, x_i^{(j)})$, $1 \leq j \leq n_i$, with $x_i^{(0)} = 0$ and $x_i^{(n_i)} = 1$, the last interval being closed. Without loss of generality, we suppose a regular division into identical length intervals and we define $\rho_i = |x_i^{(j)} - x_i^{(j-1)}| = 1/n_i$. A bucket is an elementary region of $\mathcal{E}$, result of the cartesian product of d intervals. A bucket containing more than S_{min} nodes is called a *seed*. A *cell* is then defined as an hyper-rectangle of buckets, among which there exists at most one seed. The entire coordinate space $\mathcal{E}$ is dynamically partitioned into distinct cells. Since the quality distribution is obviously not uniform over nodes, nodes are not uniformly distributed within $\mathcal{E}$. As a consequence, there are seeds containing lots of nodes, and buckets with no nodes. In addition, as network outage may impact thousands of nodes, we need to choose an architecture resilient to churn. We propose to use PeerCube [ALRB08] to organize nodes within a seed to keep the scalability property. Seeds are connected to the neighbours cell so as to provide a connected topology.

At each discrete time t, each device p reevaluates its d quality metrics. If its perceived quality has changed, it moves to its new position in FixMe. If the quality variation is perceived as an anomaly, the device indicates it in the underlying DHT representative of its quality variation. This operation allows nodes having close positions in $\mathcal{E}$ at both instants $t-1$ and t to determine the number of nodes that have perceived the same quality variation between $t-1$ and t as p experienced.

III. The Anomaly Characterization Problem

The impact of an anomaly is modeled by an anomalous movement of a group of nodes that are close to each other. Formally, given the positions of nodes at times $t-1$ and t, the set of nodes impacted by an anomaly is defined as $\mathcal{A}_t = \{j \in \mathcal{N} \mid \exists i, 1 \leq i \leq d, A_i(j,t) = \texttt{true}\}$. Moreover, given a constant R, we assume that a set of nodes $S \subseteq \mathcal{A}_t$ is impacted by the same anomaly if $\forall (j,k) \in S^2, D(j,k) \leq 2R$ with D a distance measure on $\mathcal{E}$. Finally, a set $S \subseteq \mathcal{A}_t$ of nodes is impacted by a massive anomaly if $|S| > \tau$, otherwise it is impacted by an isolated anomaly. The set of nodes impacted by a massive anomaly is denoted $\mathcal{M}_t$ and the set of nodes impacted by an isolated one is denoted $\mathcal{I}_t$.

The task of grouping a set of nodes in such a way that nodes in the same group are more similar to each other than to those in other groups is called clustering. Classical approaches [HW79], [EpKSX96] of clustering build clusters without diameter constraints. In this paper we are looking for all possible sets of diameter $2R$ containing a node p that perceived an anomaly. These sets allows us to determine if p was impacted by an massive anomaly or by an isolated one.

The **Anomaly Characterization Problem** (ACP) is defined as follows. Given a set of points $\mathcal{N} \in \mathcal{E}$, a density threshold $\tau > 1$, an anomaly radius $R \in [0,1]$, and the states of the system at instant $t-1$ and t, can we built the sets $\mathcal{M}_t$ and $\mathcal{I}_t$? Unfortunately, we can prove that this problem cannot be solved : given the positions of nodes at times $t-1$ and t, it may be possible to create two different sets of clusters where a node p belongs to $\mathcal{I}_t$ in the first set of clusters and to $\mathcal{M}_t$ in the second one. In this case it is impossible to decide whether p is impacted by a massive anomaly or by an isolated one.

We can relax the anomaly characterization problem so as to be able to solve it. We denote by $\mathcal{U}_t$ the set of nodes for which it is impossible to decide whether it is impacted by a massive anomaly or by an isolated one. The anomaly characterization relaxed problem is defined as follows. Given a set of points $\mathcal{N} \in \mathcal{E}$, a density threshold $\tau > 1$, an anomaly radius $R \in [0,1]$, and the states of the system at instant $t-1$ and t, can we built the sets $\mathcal{M}_t$, $\mathcal{I}_t$ and $\mathcal{U}_t$? We can show that it is possible to solve this relaxed version of ACP. Moreover, we exhibit local conditions which rely on the vicinity of each node that allows to solve this problem in a distributed fashion as accurately as an omniscient and centralized observer would do.

IV. Ongoing investigations and future works

As the ACP cannot be solved, it is interesting to evaluate the proportion of cases that are impossible to decide whether a node is impacted by a massive anomaly or by an isolated one. Moreover, in spite of locally computable conditions to determine if a node belongs to $\mathcal{M}_t$, $\mathcal{I}_t$ or $\mathcal{U}_t$, some of them may have a high computation cost in the worst case. We exhibit a weaker condition on the $\mathcal{M}_t$ membership. While it is cost efficient, this condition fails at characterizing all cases and thus increases the proportion of cases that fall into $\mathcal{U}_t$ class. We have to evaluate the proportion of missed detections with this condition to validate its use. The lack of available public datasets on end-to-end quality measurements lead us to built a probabilistic model to generate synthetic datasets. We are currently using it to assess the tradeoff between cost and quality of characterization in our approaches.

Finally, we are currently looking at the impact of adversarial behaviors. In the current design, nodes placement in quality space is not enforced *i.e.* nodes can lie on their quality metrics. In such a context, malicious nodes can collide in order to simulate the impact of a massive anomaly, aiming at hiding real isolated anomalies. We are working to improve the proposed overlay so as to be resilient to collusions of malicious nodes.

References

[ALRB08] E. Anceaume, R. Ludinard, A. Ravoaja, and F. Vilar Brasileiro. Peercube: A hypercube-based p2p overlay robust against collusion and churn. In *Proceedings of the IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*, pages 15–24, 2008.

[ALS+12] E. Anceaume, R. Ludinard, B. Sericola, E. Le Merrer, and G. Straub. FixMe: A Self-organizing Isolated Anomaly Detection Architecture for Large Scale Distributed Systems. In *Proceedings of the 16th International Conference On Principles Of Distributed Systems (OPODIS)*, pages 1–15, 2012.

[Bro] Broadband Forum. TR-069 CPE WAN Management Protocol Issue 1, Amend.4, 2011.

[CBG10] David R. Choffnes, Fabián E. Bustamante, and Zihui Ge. Crowdsourcing service-level network event monitoring. In *SIGCOMM*, pages 387–398, 2010.

[EpKSX96] Martin Ester, Hans peter Kriegel, Jörg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[HW79] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *JSTOR: Applied Statistics*, 28(1):100–108, 1979.

[SMK+01] Ion Stoica, Robert Morris, David R. Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, pages 149–160, 2001.