A Reliable Crisis Information System to Share Data after the Event of a Large-Scale Disaster

Marcello Cinque, Domenico Cotroneo, Christian Esposito, Mario Fiorentino, and Stefano Russo

Consorzio Interuniversitario Nazionale per l'Informatica (CINI)
Via Cinthia, Monte Sant'Angelo, 80125 - Napoli, Italy
{macinque, cotroneo, christian.esposito, sterusso}@unina.it - mario.fiorentino@consorzio-cini.it

Abstract

Crisis information systems have reliability as a key technical concern to be guaranteed, given the critical role of information sharing in crisis management. However, a common agreement is missing about the requirements to be satisfied. Moreover, scarce details are provided to describe the means to apply in order to satisfy the reliability requirements and/or to tolerate the faults that may affect a crisis information system. The scope of this paper is to determine the requirements to be satisfied in order to provide reliability, to survey the available literature and practice on this topic, and to propose an innovative solution for reliable crisis information systems in the context of the platform under development within the EU-funded DESTRIERO project.

***Index Terms*—Crisis Management Systems; Reliability; Replication; Forward Error Correction**

I. Introduction

The traditional approach of a single organization involved in the crisis management and recover is not feasible due to the increasing occurrence of large-scale disasters; but, several different first responder organizations at national and international level must act together, and may be coordinated by a given organization such as OCHA (Office for the Coordination of Humanitarian Affairs), paving the way for a Collaborative Crisis Management (CCM). Since decision-making must be flexible and responsive[1], there is a great demand for CCM intensive information sharing among the organizations involved in the crisis management. However, we can notice a lack of proper technological solutions, which enable cross-border networks of crisis management organizations to set-up flexible support systems for a joint operation in which information is continuously updated and shared between organizations, leveraging on the Information Systems existing within the involved organizations, and in which progress is monitored and resource sharing is facilitated. DESTRIERO is an EU-funded project that aims at providing an answer to this lack by designing and developing a middleware platform for crisis information systems as a mean to provide data sharing and cooperation capabilities to integrate heterogeneous systems and information sources and to support damage and needs assessment as well as recovery planning.

In this context, reliability becomes a key technical concern, given the critical role of information sharing in crisis management. In this paper, we identify the most suitable reliability solutions for a crisis information system as the one under development in DESTRIERO. Since there is no common agreement on the requirements to be satisfied in order to assume that a crisis information system is reliable, in Section II we start by reporting the collection of reliability requirements from different end-users and technology experts in the domain of crisis management, and then, we analyze the available literature on the topic. Knowing what is lacking in the current literature and practice, we propose in Section III a set of means to tolerate the faults that may occur during the life of a crisis information system with reaching an invalid state and compromising the mission of the system. Section IV presents empirical results proving the validity of our approach. We conclude with final remarks in Section V.

II. Requirements and State of the Art

A widely accepted definition of reliability conceives it as the continuity of correct service, despite of the occurrence of possible faults that may compromise the correct behavior of the system. In some papers dealing with

[1] http://www.oecd.org/futures/globalprospects/40867519.pdf

crisis information systems, we have also found references to data reliability, as the ability of an information system in handling accurate, trustworthy and honest piece of information about a crisis event [1]. Such a kind of reliability requires means to verify accuracy and reputation in data acquisition and processing. In this document, we will refer to the first reliability definition. A term that is frequently coupled with reliability in articles and vendor documents is resiliency, *i.e.*, the ability to adapt under stress or faults in order to avoid failure and to continue to offer some level of lowly-worsened performance [2]. The difference is that a reliable system is essentially one that functions as the designer intended it to, when it is expected to, and wherever the customer is connected; while, a resilient system is able to withstand certain types of failure and yet remain functional from the customer perspective. In the literature of crisis management, there are some references to societal resilience [3] where citizen, first-responders and operational commanders are trained to act efficiently and independently in crisis events based on a set of core values, ethics and priorities to guide them in their decisions and actions after a disaster. The intentions of this work is not to address such a broad concept, but to build the bases to create resilient data sharing networks able to have governance frameworks that enhance societal resilience.

In order to determine the requirements that make a crisis information system resilient, and therefore, reliable, we have to determine a fault model for this kind of systems, *i.e.*, identifying what kinds of faults may occur and compromise the overall system. A crisis information system is composed of several nodes, interconnected by communication channels. Both processes and channels may expose faulty behaviors, which can be modeled as Interruptions or Crashes [4] with processes and links suddenly stopping to work, *i.e.*, nodes stop to produce data or react to incoming data and links experience complete loss of connectivity. Even when links are working properly, the network of interconnected routers and nodes may lead to several communication faults, such as (i) Data Loss, (ii) Unexpected Delays, (iii) Message Corruption, and (iv) Network Partitioning.

Given such a fault model, the required fault-tolerance means needed by a crisis information system can be summarized as follows [4]. First, to handle losses and incorrect network behavior, proper recovery means are needed so as to assure that messages are delivered to the intended destinations at the right time. Second, to deal with interruptions, the design of the system, at any of its levels, has to assume a proper degree of redundancy, so that in case of the unavailability of a node or link, another is ready to take its place. Third, a proper logging system is required to monitor that the crisis information system is behaving as expected, and to raise alarms in case of deviations in order to apply proper mitigation actions able to bring the system in a safe state and in a correct behavior. Last, crisis management always requires that decisions must be taken at real-time so as to timely and promptly adopt the needed countermeasures to face consequences of a disaster and to reduce the number of causalities and injuries. This also means that fault-tolerance has to be timely and does not have to affect the performance of the system.

In the current literature and practice on crisis information systems, there is scarce information on fault tolerance. The reason is that the research community has focused on different topics, and reliability concerns are left to the middleware technology used for the integration of the different parts of a crisis information system. Practical examples are available in the literature where crisis information systems are built on reliable and resilient networking and the other system reliability requirements are neglected. In fact, having resilient communication channels working during crisis for local blue light services is necessary to have response preparedness during a crisis of emergency services. However, it is not sufficient since it is crucial to also achieve a high degree of reliability for the various kinds of data required to make timely decisions during crisis management (*i.e.*, reliability of the data storage layer), for the various data processing engines able to transform and aggregate the required data (*i.e.*, reliability of the data processing layer), and /or for the different visualization operations (*i.e.*, reliability of the data visualization layer).

III. Proposed Approach

Without loss of generality, a large-scale crisis information system is the integration of different systems by means of large-scale networks. In each system, we can find a series of nodes belonging to a first responder's organization and a DESTRIERO node, which hosts the functionalities of the DESTRIERO platform made available to the services and operators of the hosting organization. Each time the information is delivered to a DESTRIERO node, it may distribute locally to the other nodes, considering the needs of such nodes and providing an illusion that the information has been internally produced and not received from another organization. In order to satisfy the reliability requirements mentioned above the information of interest should not be hold only within a single organization, but distributed among the organizations so that in case of a failure, it is still available within the overall infrastructure and its access latency is optimized since it can be retrieved from a more convenient location than the crashed one where it has been produced. Moreover, a proper resilient multicast is achieved so that the information of interest is guaranteed to reach the intended destinations. We leave as a future work the design of a logging scheme to monitor

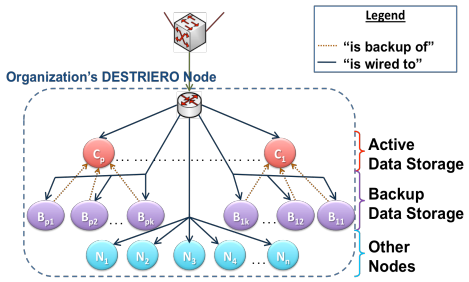


Fig. 1: Replicated data storage within the DESTRIERO node.

the correct behavior of the system and trigger possible maintenance actions to recover from a faulty state.

A. Replication Scheme

A well-known solution to tolerate the crash problem is to replicate the entity for which it is crucial to tolerate its eventual crash. In the literature, we can find two different techniques: passive replication scheme, where the failed entity is replaced by one of its backups; and active replication , in which the system exposes a virtual entity made up of a set of distinct replicated objects with equal responsibilities. The first one presents the issue that the entity would be down for a certain time window, in the second one there is still the possibility of a failure of all the coordinators due to common mode errors. Hybrid schemes are possible, such as semi-active replication [5], where the entity is actively p-redundant, *i.e.*, there are p replicated objects active at the same time; moreover, there are k backups for each active object. Such a replication scheme allows the system to tolerate entity crashes without having certain functionalities unavailable for a certain period of time, without being vulnerable to common mode errors.

In our work, the application of a redundancy degree within a typical large-scale crisis information system is realized by adopting the above introduced hybrid model. Specifically, we have considered as critical the components in each first responders' organizations that composes the data storage layer, since the loss of such amount of data can be catastrophic for the overall system. Therefore, the data storage within the DESTRIERO node deployed at the premises of an organization is schematically structured as a set of nodes, each identified by a proper string and interconnected as in Figure 1. A set of nodes for data storage is running and receiving all the incoming requests arrived to the organization. The write requests, which are the ones that alter the data stored in these nodes, are executed by all the nodes so as to keep consistent their state among each other, *i.e.*, all detain the same pieces of data at the same version and content. At the conclusion of each write request, the node logs an entry in a proper file that keeps the history of past activities for the node. Such a mechanism is required so as to undo any possible change in any time and to know who is the responsible for each change. The read requests, which aim to return a copy of certain stored data to the requesting organization, are executed only by one of the active data storage nodes, *i.e.*, the first available one. This allows evading the incoming read requests in parallel and reducing the overall retrieval latency. Moreover, due to nodes capabilities to synchronize themselves, read requests can be executed in parallel among each other and in series with a write request, which is pre-emptive with respect to the reads. A set of nodes for data storage is running and assumed to be the backup of one of the active nodes in the previous set. Such nodes receive only the writing requests, and change their internal state, *i.e.*, the stored data, accordingly. This is needed to have the backups consistent with the active nodes. Whenever an active node is compromised, it is substituted by one of the available backups, *i.e.*, the node with the greater identifier is the new active node.

We can assume two kinds of overlays, the one within each organization that interconnects the nodes storing data, as illustrated in Figure 1 with active and backup nodes, and an additional level with a single active node per organization being interconnected with its peers in the other organizations. Such a node is called super-peer and has the duty of determining which data is needed to be kept by its organization of reference. The other active nodes behave as backup in case the super-peer may result unavailable. When a super-peer is detected as unavailable, by means of keep-alive messages or timeout-base mechanism, the replica with the higher identifier is automatically elected as the super-peer of the organization. When data is replicated, write and read requests are not only directed towards the organization that has produced the data of interest, but all the organizations that hold in their data storage system such a data.

To help finding such information, the data available in the platform is classified in different types by placing a special string indicating such a classification, and the super-peers, and their replicas, holds a map that associate each class of data to the address of the different nodes holding a replica of such data. The remaining issue left to be solved is how to determine where to place replicas of each data class. Such a decision is taken without the intervention of any centralized managers, but the super-peers collaborate among each other to find a solution to this problem in a distributed manner. Specifically, each super-peer keeps a set of statistics for each class of data requested by any entity in its organization. We have envisioned two classes of statistics: data availability and retrieval time. In the first case, the super-peers counts the number of successful write/read request over the total

number of requests, in the second case, computes the average time to retrieve instances of the given class of data. Periodically, the super-peer computes a satisfaction degree per each own i-th organization knowing the number of available replicas and their placement within the overall DESTRIERO platform:

$$\delta_i(n, \overline{P}) = \omega_A \left(\rho_A - \frac{\kappa_{Succ}}{\kappa_{Tot}} \right) + \omega_\Lambda - \kappa_\Lambda \qquad (1)$$

where ω_A and ω_Λ are weights chosen by organization administrator in the $[0, 1]$ interval and stating the importance of availability over latency, κ_{Succ} is the total number of successful requests, κ_{Tot} is the number of requests, κ_Λ is the computed average retrieval latency, ρ_A and ρ_Λ are the required level of data availability and retrieval latency chosen per each class of data by the organization administrator. If the satisfaction degree is positive, it means that the current setting of the platform satisfies the requirements of the users of the given organization and no further actions are needed. On the contrary, a negative value states that the current setting is not satisfactory and changes are required. The costs for an organization for holding a replica for the given class of data can be formulated as follows:

$$cost_i = \omega_M \left(\frac{\theta_{data}}{\theta_{Tot}} \cdot 100 \right) + \omega_\Lambda \left(\frac{\iota_{data}}{\iota_{Tot}} \cdot 100 \right) \qquad (2)$$

where the first contribution measure the fraction of the storage resources used to hold the replica, while the second one indicates the fraction of requests received for the hold replica over the total number of served requests. The driving idea is to maximize the satisfaction degree at each organization and minimizing the relative costs by determining an optimal number and placement of replicas for the class of data of interest:

$$\begin{array}{l} \max_{i \in [0,N]} \delta_i(n, \overline{P}) \\ \min_{i \in [0,N]} cost_i \cdot P_i \end{array} \qquad (3)$$

subject to:

$$\begin{array}{l} n = \sum_{i=0}^{N} P_i \leq max \leq N \\ P_i = \begin{cases} 0 & \text{if no replica at the i-th organization} \\ 1 & \text{if replica at the i-th organization} \end{cases} \end{array} \qquad (4)$$

where max indicates the maximum number of replicas to be placed in the infrastructure made of N organizations integrated by the DESTRIERO platform. This is an example of multi-objective optimization problem, since the two objective functions to satisfy are opposing, *i.e.*, a solution for the first maximization is not a solution for the second minimization, and a trade-off is needed. Such a trade-off is called Pareto solution, *i.e.*, a configuration able to keep in balance the two objectives and a different solution is not able to improve such a situation. Classical solutions [6] are not viable in our case, since the DESTRIERO platform is large and global knowledge in such systems is not achievable. Therefore, we have considered a distributed approach to the resolution of this problem by means of a non-cooperative game [7].

More formally, let us consider the super-peer overlay as an undirected graph (X, E), consisting in a set of nodes, namely X, and a set of edges connecting two nodes, namely E. Within the set of the available nodes, we define a subset of all the nodes, namely $Y \subset X$, containing the nodes where a player can be located. We consider a set of players $P := \{c1, c2, \cdots, cp\}$ of finite size $p \leq 2$. Formally, the strategy set for each player $c \in P$ is defined as $S^c = Y$, such that a strategy of a player is the selection of a node $s^c \in Y$. Combining the strategy sets of all the players, namely $S = S^{(c_1)} \times S^{(c_2)} \times \cdots \times S^{(c_p)}$, a strategy profile $s \in S$ implies a certain payoff to each player c, namely $\Phi^{c(s)}$, which are aggregated in the so-called profile of payoffs denoted as Φ^S. The payoff is the gain achievable by a player to host a given replica considering the store the replica and to manage the incoming requests to read/write it, as mentioned above.

The scope of the game is to determine the best strategy profile that implies the maximum payoff for all the players. Despite the several possible formulations that came out within the literature, we have described such a game in terms of a non-cooperative game, where players are selfish, *i.e.*, there is no direct communication between the players, and each one only cares to maximize its own profit or to minimize its own costs without considering the state of the other players (with the eventuality of damaging them, even if it is not intentional). Then, the normal form for the non-cooperative game for our replica placement problem is given by $\Gamma = (P, S, \pi)$, with the objective of maximizing the payoff for all the players, for which we are interested in finding Nash equilibria, *i.e.*, given a certain strategy $s \in S$, it is not profitable for a player to select a different replica placement pattern than the one in the current strategy profile since adding or removing a replica will not change or even reduce the achievable payoff, so a player has no incentive to change strategy. The demonstration of the existence of such equilibria is a known NP-hard problem and is resolved by means of theorems. For a concrete example, the authors in [8] demonstrate the existence of at least a Nash equilibrium for games as ours and the conditions to induce such equilibrium are presented.

In our game, a strategy is represented by a binary decision to hold a replica or not, and we consider the costs for each player, according to the previous formulation of $cost_i$, properly assigned to each player. Specifically, let us indicate with S_i the binary value representing the strategy chosen by the i-th player (which is 1 if the i-th player decides to hold a replica; otherwise, it is 0). The cost paid by the i-th player to follow its strategy can be formalized

as follows:

$$C_i(S_i) = cost_i \cdot S_i + \delta_i \cdot (1 - S_i) \quad (5)$$

where $cost_i$ and δ_i are expressed in the previous equations. The game can start with a random strategy profile and evolve over the time where each player changes its strategy so as to minimize its costs formulated in the previous equation. Such an evolution will bring to a stable solution represented by the Nash Equilibrium, where no player has an incentive to change its strategy. Based on the definition of a Nash Equilibrium, it is possible to see that a strategy profile s represents a Nash Equilibrium if and only if the two following conditions are guaranteed:

$$\exists i \in Y s.t.\ \delta_i \leq cost_i \quad (6)$$

and

$$\nexists i \in Y s.t.\ cost_i - \delta_i > 0 \quad (7)$$

The first condition indicates that the satisfaction generated by a replica placed at the i-th node is never greater than the cost of placing it; so, none of neighboring nodes has an incentive to act as a codec. While, the second condition states that, when the i-th node holds a replica, it is not convenient to stop holding it since the paid cost is already minimized. The condition in the first equation defines the control behavior of the super-peers to decide holding a replica or not.

B. Resilient Multicasting

Communications among DESTRIERO nodes in different organizations are conveyed by the Internet, so they are affected by link crashes and bursty loss patterns. In addition, due to the introduction of the replication scheme in the platform, the data exchange pattern adopted within the system is a multicast one. The general literature of reliability means to tolerate losses in a multicast communication infrastructure can be roughly classified in two big classes [9]: the one based on temporal redundancy, and the one based on spatial redundancy. Since in crisis information systems performance matters, we have focused our attention on spatial redundancy, among which the main ones are Forward Error Correction (FEC), and Path Redundancy, both affected by several drawbacks, as described in [9]. We have decided to apply a hybrid approach by properly combining FEC and Path Redundancy into a proper communication protocol. At this aim, we have applied a protocol we have designed in [10] to build multiple diverse trees by selecting the paths from any new node to its parents that expose the lowest measure of diversity, and keeping the paths to the children of a given a parent to maintain a measure of diversity closer to the value they had before the inclusion of the new node as a child.

Disseminating over multiple trees is not so effective when the dissemination infrastructure is affected by losses. To lower such a probability, redundant packets are generate by using a FEC technique, and to forward a portion of the encoded packets per each tree. The producer of a message applies a coding technique to generate r redundant packets from the k information packets (so that the packets to deliver to each node is $n = k + r$). Then, it equally disseminates the n packets though the t multiple trees (*i.e.*, each tree conveys a number of packets equal to n/t). If there may happen only a single link crash that compromises the message delivery along a single tree, each node will receive only $n - n/t$ packets, so the system will tolerate the crash if the number of received messages is greater or equal to the capacity, namely C of the adopted coding technique:

$$n - \frac{n}{t} \leq C \rightarrow \left(1 - \frac{1}{t}\right) \cdot n \geq C \rightarrow \left(1 - \frac{1}{t}\right) \cdot (k+r) \geq C \quad (8)$$

Considering this equation, we can formulate a condition on the applied redundancy degree r, based the size in packets of the message to be sent, namely k, so that a single link crash can be tolerated:

$$r \geq \left(1 - \frac{1}{t}\right) \cdot (C - k) \quad (9)$$

This result can be generalized for the number, namely ft, of faulty trees, *i.e.*, they do not deliver their packets to a certain node due to a link crash, as follows:

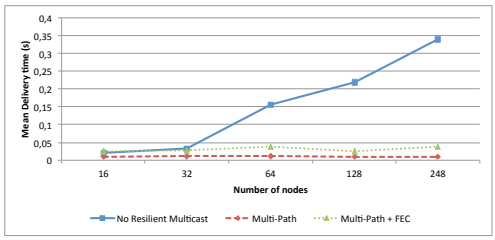
$$r \geq \left(1 - \frac{1}{ft}\right) \cdot (C - k) \qquad \text{iff } ft < t - 1 \quad (10)$$

Such a consideration on link crashes can be applied also to message losses, but in this case not only the message producer can apply FEC, but all the nodes in the multi forest, so as to protect each of its links from losses.

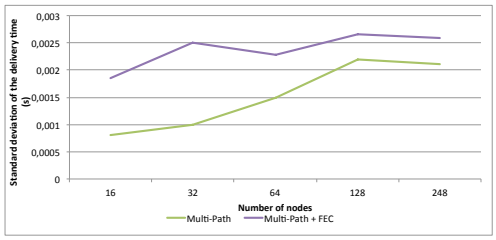
IV. Empirical Evaluation

We have implemented our solution by using the OMNET++[2] simulator and conducted several simulations. In our first ones, the exchanged messages have a size of 23 KB, the publication rate is one message per second and the total number of nodes is 40. The network behaviour has 50 ms as link delay, and 0.02 as loss rate. The coding and decoding time are respectively equal to 5ms and 10ms. In the second simulations, we assume that each organization, made of 4 nodes, periodically generate a datum of 23 KB, while others periodically make requests for a piece of data previously produced by one of the remaining organizations. We assume that nodes can crashes with a probability of 0.05.

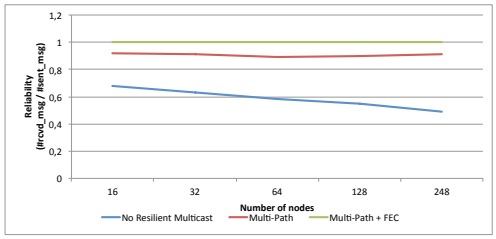
[2] www.omnetpp.org



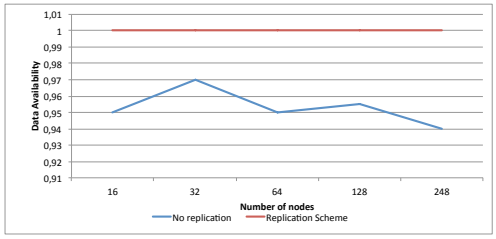
(a)



(b)



(c)



(d)

Fig. 2: Experimental results.

Figure 2(c) shows that the multiple-tree solution alone is able to increase the success rate; however, it is not able to reach full success, which is achieved when FEC is also used. These solutions are not only able to augment the achievable success rate, but also to reduce the delivery delay due to their intrinsic parallelism in the data dissemination, as proved by Figure 2(a). Both solutions exhibit stable performances demonstrated by very low standard deviation in the delivery delay, as illustrated in Figure 2(b). Data availability is quite low without any replication, as shown in Figure 2(d), while our replication approach is able to tolerate node crashes without compromising the data availability. For space limit, we are not able to show also the performance of our solution, which has proved to speed up in the data retrieval operation equal to 35% than the case without any replication.

V. Final Remarks

This paper describes to make a Crisis Information System reliable by using (i) a replication scheme to deal with the possible interruptions of nodes and network partitions, and (ii) a resilient multicasting to cope with network failures and crashes. We have identified as possible future work the design of a logging strategy in order to detect possible failures and determine their causes.

Acknowledge

This work has been partially supported by the EC in the framework of the Collaborative Project “A DEcision Support Tool for Reconstruction and recovery and for the IntEroperability of international Relief units in case Of complex crises situations, including CBRN contamination risks” (DESTRIERO, http://www.destriero-fp7.eu/ - Grant agreement no: 312721).

References

[1] B. Ley and V. Pipek and C. Reuter and Torben Wiedenhoefer . Supporting improvisation work in inter-organizational crisis management. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.

[2] Y.Y. Haimes. On the Definition of Resilience in Systems. *Risk Analysis*, 29(4):498–501, 2011.

[3] A. Boin, and Allan McConnell. Preparing for Critical Infrastructure Breakdowns: The Limits of Crisis management and the Need for Resilience. *Journal of Contingencies and Crisis Management*, 15(1):50–59, 2007.

[4] M. Cinque and C. Di Martino and C. Esposito . On data dissemination for large-scale complex critical infrastructures. *Computer Networks*, 56(4):1215–1235, 2012.

[5] X. Defago and A. Schiper and N. Sergent. Semi-passive replication. *Proceedings of the Seventeenth IEEE Symposium on Reliable Distributed Systems*, pages 43–50, 1998.

[6] D.F. Jones and S.K. Mirrazavi and M. Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137(1):1–9, 2002.

[7] J. Cardinal and M. Hoefer. Non-cooperative facility location and covering games. *Theoretical Computer Science*, 411(16-18):1855–1876, 2010.

[8] A. Vetta. Nash equilibria in competitive societies, with applications to facility location, traffic routing and auctions. *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, pages 416–425, 2002.

[9] C. Esposito, D. Cotroneo, and S. Russo. On reliability in publish/subscribe services. *Computer Networks*, 57(5):1318–1343, 2013.

[10] C. Esposito, D. Cotroneo, and A. Gokhale. Reliable publish/subscribe middleware for time-sensitive internet-scale applications. *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems (DEBS)*, pages 16:1–16:12, 2009.