Self-adapting Strategies guided by Diagnosis and Situation Assessment in Collaborative Communicating Systems

A. Subias [1], E. Exposito [1], C. Chassot [1], L. Travé-Massuyes [1], K. Drira [1]

[1] *CNRS; LAAS; 7, avenue du Colonel Roche, F-31077 Toulouse, France*
Université de Toulouse; UPS, INSA, INP, ISAE; LAAS; F-31077 Toulouse, France

subias,exposito,chassot,louise,drira@laas.fr

ABSTRACT

Coping with context changes in networked systems requires considering self-adaptive communication protocols in the design of the future generation of communication systems. The communication system configuration then dynamically changes according to the user's requirements and to the load of the communication resources. Dealing with this problem requires the capacity of detecting the possible degradations of the Quality of Service (QoS) and of dynamically modifying the behavior of the communication protocols for each new context situation. This requires in turn both monitoring the QoS values, detecting the degradations, identifying their origins through appropriate diagnosis and executing reconfiguration actions. We propose to implement such functions by considering an event-based model-driven diagnosis approach leading the dynamic composition of communication protocols for the execution of the reconfiguration. We consider a chronicle-based diagnosis approach and we apply our ideas to the Transport layer of communication systems. In this position paper, we show the relevance of our ideas and present preliminary ideas towards an integrated automated approach.

1 INTRODUCTION

More and more systems take benefit of communication supports and achieve their objectives in a networked distributed and cooperative way. Building networked systems composed of instrumented and intelligent artifacts, hence equipped with processors, sensors and actuators has became a requirement for numerous applications. Think for instance of PDA (Personal Digital Assistant)-like and smart phone mobile equipments connected through wireless ad-hoc networks, or mobile robot floats organized towards a cooperative goal and you get a flavor of the targeted technologies and applications. These systems and environments rise interdisciplinary scientific challenges at all levels and call for novel methods.
Among the challenges, this paper focuses on the problem of providing adaptability to the traffic control and management system. More precisely, we are interested in adapting the communication protocols at the transport level for coping with the dynamically changing context situations arising from the distributed and collaborative mobile applications.
Starting from initial works described in (VanWambeke *et al.*, 2007) (VanWambeke *et al.*, 2008), two ways of tackling this problem are foreseen in the paper. The first one is to orchestrate the traffic with a composition of micro-protocols that provide different and well-identified QoS properties to the traffic. The second one proceeds from a multiple connections point of view and acts through collaborative management of the underlying transport protocols involving different local and remote coordination entities.
Standard transport protocols implement strategies that act upon hard-coded decisions based on rather simple traffic situation assessment (e.g. reception of duplicate acknowledgments). Our proposal builds on works by the diagnosis community and suggests the use of more sophisticated dynamic models associated to the different relevant traffic situations to be detected to guide the reconfiguration strategies. We show that the discriminating features of such situations can be accounted by the chronicle formalism, well-known in the event model based diagnosis domain and particularly suited to on-line recognition (Dousson *et al.*, 1993). Chronicles can also be used for diagnosability analysis when checked for exclusiveness, hence providing a way to infer new relevant traffic features to be monitored (Pencolé and Subias, 2009). Our approach hence contributes to elaborating a model-based diagnosis approach for the correct design of adaptive communication protocol. We identify two problems that we handle through two appropriately selected case studies to illustrate our proposal: the initial choice and the continuous adaptation of the micro-protocols composition for managing the end-to-end TCP-like[1] classical transport connection, what we call *intra-connection strategy*, and the collaborative management of multiple connections called *inter-connection strategy*.

The paper is organised as follows. Section 2 provides a state of the art of the large diversity of services currently offered by the transport layer and further motivates the work. Section 3 presents the chronicle formalism and its suitability to the addressed problem. Section 4.1 describes an intra-connection scenario and illustrates chronicle relevancy. Section 4.2 focuses on

[1] TCP is the Internet standard for transport protocols.

the inter-connection problem and draws the lines of situation recognition for network load balancing. Finally, a concluding section summarizes the paper and outlines the topics for future work.

2 TRANSPORT PROTOCOLS TAXONOMY

The well-known Open System Interconnection (OSI) model (Zimmermann, 1980) describes how information from an application in one computer moves through a communication medium to an application in another computer. It represents a referential model composed of seven layers, each one specifying particular communication functions. The transport layer is the lowest layer operating on an end-to-end basis between two or more communicating hosts. This layer is located between the applications and the network layer. A "service" is defined as the abstraction capability offered by an OSI layer to its higher layer. Transport services enable applications to abstract the communication services and protocols provided by the lower network and MAC (Media Access Control) layers. In contrast, transport protocols specify the mechanisms to be implemented in order to offer the required transport services.

Basic transport services are characterized by:

- Connection-oriented vs. connection-less. A transport user (i.e. two communicating application-level entities) generally performs three phases when using transport service: connection establishment, data transfer and connection termination. A connection-oriented transport service provides primitives for the three operations; a connection-less service supports only data transfer primitives. Moreover, a connection-oriented service maintains state information about the connection (i.e. message sequence number, buffer sizes, etc).
- Message-oriented vs. byte-stream oriented. Transport services offer two transfer modes: message oriented and byte-stream oriented. In the former, users send messages or service data units ($SDUs$) having a specified maximum size and message boundaries are preserved. In the byte-stream service the data sent by the user is transported as a flow of bytes and messages boundaries are not preserved.
- Reliability and order. A transport service offering a fully reliable and fully ordered service guarantees no-loss, no-duplicates, ordered and data integrity in the user data delivery.
- Multicast vs. unicast. A multicast service enables a sender to deliver data to one or more receivers. A unicast service limits the data delivery to exactly one user receiver.
- Quality of service. A transport layer that explicitly provides QoS allows a sender to specify the quality of the transmission service required (i.e. classically: delay, jitter, order, reliability and throughput). These would be the basic parameters guiding the transport classification presented in this work.

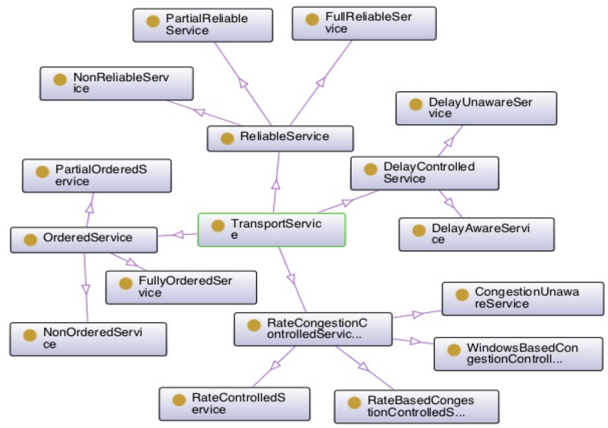

Figure 1: Transport Services Characterization

2.1 Transport service characterization

Among the most well-known transport protocols, we can mention TCP (Transmission Control Protocol), UDP (User Datagram Protocol), $SCTP$ (Stream Control Transmission Protocol), $DCCP$ (Datagram Congestion Control Protocol) and $MPTCP$ (Multipath Transport Protocol).

A characterization of transport services is now presented based on the following set of categories (see Figure 1):

- Reliable services: Fully Reliable, Partially Reliable or Unreliable;
- Ordered services: Fully Ordered, Partially Ordered or Non ordered;
- Rate/congestion controlled services: Windows-Based, Rate-Based or Congestion-Unaware;
- Delay controlled services: Delay-Aware, Delay-Unaware.

Based on this hierarchy, a taxonomy of transport services provided by the existing transport protocols has been elaborated. Individuals of this taxonomy are the following:

- TCP: Fully Reliable, Fully Ordered, Windows-Based Congestion Controlled, Delay-Unaware;
- UDP: Non Reliable, Non Ordered, Congestion-Unaware, Delay-Unaware;
- $SCTP$: Fully Reliable, Partially Ordered, Windows-Based Congestion Controlled, Delay-Unaware;
- $DCCP$: Non Reliable, Non Ordered, Delay-Unaware and Congestion Controlled; $DCCP$ offers several congestion control instances, in our cased identified as DCCP-2 for a Rate-Based Congestion Control and DCCP-3 for a Windows-Based Congestion Control. Figure 2 illustrates this transport service taxonomy.

2.2 Transport service limitations

The previous taxonomy shows that existing transport protocols present several limitations with regard to

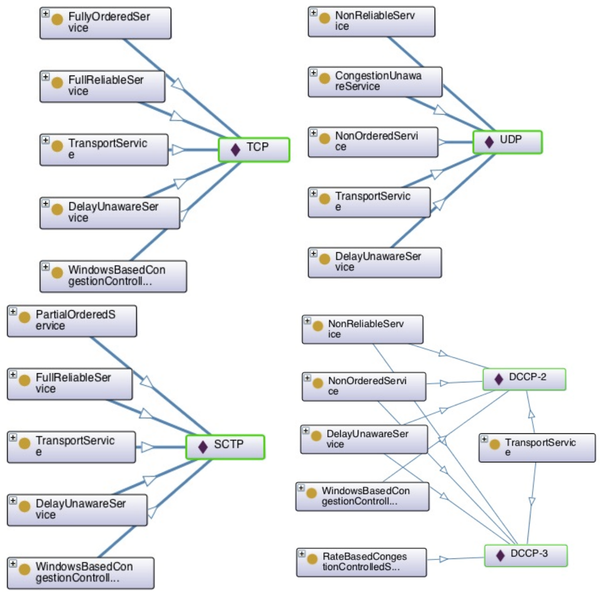

Figure 2: Transport Services Taxonomy

QoS application requirements. For instance, no protocol takes into account requirements in terms of delay (all are Delay-Unaware). Likewise, partially reliable and partially ordered services, more compliant with several kinds of video/audio multimedia applications, are not provided. However, mechanisms able to take benefit of packet losses and out of order tolerance of applications could be developed to provide delay controlled services. Moreover, most of the previously presented transport services are based on implementations where mechanisms offering different functionalities (i.e. error control or congestion control) are merged within the same monolithic implementation. Such a solution has a limited scope of applicability. It assumes a predefined QoS handling mechanisms already known and integrated during the design-time of the communication protocol. A component-based approach such as the one proposed by $MPTCP$ and generalized by ETP (Enhanced Transport Protocol) (VanWambeke *et al.*, 2007) could widely facilitate the design and development of new composed transport services. Indeed, new transport services could result of the combination of pluggable components offering specific functionalities. The problem considered in this paper is to characterize from the network available data the situations that call for specific basic functionalities and specific compositions.

3 TAKING BENEFIT OF SITUATION RECOGNITION

The problem of having an explicit assessment of the communication situation is similar to a monitoring problem as considered in the diagnosis community. Standard transport protocols implement strategies that act upon simple hard-coded event patterns representing specific traffic situations. These are detected at the source or destination nodes. The events arise from feedback provided by standard parameters stamping the packets and the event patterns generally express temporal relations. Our claim is that we can formalize the temporal patterns in use in standard protocols using *chronicles* with the aim to express more sophisticated patterns and hence to generalize traffic situation assessment.

The section below presents chronicle recognition and provides a state of the art. Next section illustrates the feasability of the approach by providing a chronicle model of a standard event pattern used by TFRC.

3.1 Chronicles recognition

Chronicle recognition is a commonly used method to capture automatically the evolutions or partial evolutions of dynamic systems. The evolutions to monitor are described in terms of temporal patterns called *chronicles*. For instance one may want to track dynamic situations like the ones below:

- an event A is immediately followed by an event B,
- an event A is followed by events B and C
- an event A is immediately followed by an event B without an event C between A et B
- an event A is followed by events B and C in less than 10 time units
- an event B occurs after 8 occurrences of an event A
- ...

This kind of approach assumes that a time stamp or occurrence date can be assigned to each event. A chronicle is then the description of a temporal pattern in terms of events and time relations i.e. time constraints between event occurrence dates. Time constraints are durations expressed as bounds on the difference between two time points or event occurrence dates.

Chronicle recognition consists in identifying in an observable flow of events all the instances of the chronicles. The identification is performed on the fly, as soon as the events occur. This approach is very efficient for situation recognition as it relies on the direct link between the symptom of a situation and the situation itself. Nevertheless, it differs from classical abductive diagnosis systems as time aspects are dominant.

Chronicle based approaches can be related to others methods to represent situations stressing on the temporal dimension such that situation calculus introduced by (McCarthy and Hayes, 1969), the event calculus (Kowalski and Sergot, 1986) or the temporal interval of Allen (Allen, 1983),(Allen, 1984). All this methods are commonly used in the Artificial Intelligence field for representing and reasoning about temporal information. One major advantage of chronicles compared to these approaches is the rich formalism allowing one to describe the observable patterns corresponding to behaviors one wants to detect.

In particular, chronicles account for partial orders between events easily and are also able to express forbidden events i.e. the lack of events. Indeed, forbidden events are events that must not occur during the recognition. A chronicle could be defined for instance by

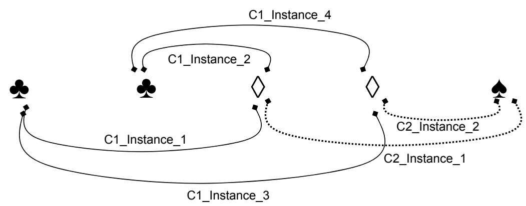

Figure 3: Instances of chronicles

"an event B following an event A, without an event C between A and B.

Another advantage lies on the efficiency of the recognition system which makes chronicles suitable for real-time operation. The chronicle recognition system aims to identify all possible matchings between an input flow of events and a chronicle. When a new event occurs it is integrated into the chronicle if it is consistent with the expected event of the pattern and if its time stamp is consistent with the time constraints of the chronicle. Each new instance of chronicle generated is a new hypothesis and added to the set of hypotheses. The chronicle recognition system must then manage on-line all these instances i.e. all the hypotheses elaborated in time. Instances are discarded when time constraints are violated. A chronicle is recognized when a complete match is observed. For one given flow of events multiple instances of a chronicle can be recognized in a sequential way or simultaneously. For example, let us consider the two simple chronicles:

1. C_1: event ♣ followed by event ♢
2. C_2: event ♢ followed by event ♠.

Figure 3 gives all the instances of the two chronicles for a given observed event flow: ♣ ♣ ♢ ♢. Four instances of C_1 can be identified. $C_1\_instance_1$ and $C_1\_instance_3$ are recognized sequentially whereas $C_1\_instance_1$ and $C_1\_instance_2$ are recognized simultaneously. If the event flow is extended by the event ♠ two instances of C_2 are also identified. Note that, the possible growing up of the set hypotheses is a real problem that must be controlled.

3.2 State of the art on chronicles

Most of the works on chronicles are issued from the French community The chronicle notion was developed in (Dousson *et al.*, 1993) leading to a Chronicle Recognition System (CRS http://crs.elibel.tm.fr). In this approach the chronicles are expressed in a specific language and then translated into temporal constraint satisfaction graphs. This chronicle approach has been developed and used in several applications (Cordier and Dousson, 2000): in the medical field for ECG interpretation and cardiac arrhythmia detection (Carrault *et al.*, 1999). It is also used in intrusion detection system (Morin and Debar, 2003) or in telecommunication systems (Laborie and Krivine, 1997). It has also been used for human security purposes to detect suspect human behavior operators (Rota and Thonnat, 2000). More recently chronicles have been used in the context of web services (Cordier *et al.*, 2007)(Pencolé and Subias, 2009). Another chronicle based approach has been developed by (Carle *et al.*, 1998) providing also a chronicle recognition system called $CRS/ONERA$ able to detect on-line chronicle instances. This approach has been used for intrusion detection, for human machine interface and also for behavior analysis in the context of HLA simulations (Bertrand *et al.*, 2008).

The main drawback of the chronicle based approach is the difficulty of acquiring and updating the chronicle base. To remedy this problem learning techniques have been proposed (Mayer, 1998). Another solution is based on analyzing logs and extracting the significant patterns by data mining techniques (Dousson and Duong, 1999). In (Guerraz and Dousson, 2004) the patterns are built from the fault models using Petri nets.

4 CASE STUDIES: INTRA AND INTER CONNECTION STRATEGIES

To illustrate our proposal, we consider the context of one application involving one or several collaborative or non collaborative connections. This application is supposed to be distributed between two hosts having one or several physical network interfaces, this enables multi-path communication (by default only one path is used). It is also assumed that two types of QoS transport agreements have been established between the application and the underlying communication system: mandatory or best-effort. The former means that the required QoS must be guaranteed; the latter means that the QoS has to be maintained at the required level as best as possible.

The purpose of the case studies presented in this section is to explore the use of chronicles with the aim to perform an autonomous transport service management able to select the most adequate transport mechanisms with regard to the recognized situation. To facilitate the understanding, a limited set of transport mechanisms is considered:

- $TFRC$ or TCP Friendly Rate Control: implements a rate-based congestion control mechanism based on the observed network conditions (i.e. loss rates and round-trip-time delay);
- $TD-TFRC$ or Time-constrained and Differentiated $TFRC$: specializes the previous mechanism by taking into account the application QoS requirements expressed in terms of loss tolerance and delay constraint. TD-TFRC performs a flow adaptation by selective discarding application data when the accumulated delay is not acceptable. This selective data discarding takes into account the loss tolerance of the application;
- $MP-TFRC$ or Multipath $TFRC$: follows MPTCP's approach but applied to $TFRC$ instead of TCP.

4.1 Case study 1: Intra-connection strategies

Intra-connection strategies concern unicast flows operating such as an end-to-end TCP-like classical transport connection. In such cases, some transport pro-

```
chronicle example {
   event(e2,t2);
   event(e3,t3);
   t3-t2 <= 7
}
```

Figure 4: Basic Chronicle

```
Message  packet [ ?num_loss, ?seqn]{
         }

Chronicle loss  [?num_loss]
    {
      occurs ((3,3), packet [ ?num_loss,?seqn], (t1,t2))
       ?seqn>?num_loss
      t1<t2
      when recognized {
            emit event(packet_loss,t2);
      }
  }
```

Figure 5: Chronicle for the loss of packet

tocols implement flow and congestion control mechanisms in order to avoid exceeding receiver buffer capacities and to react to network congestions. The principle of the congestion control mechanism is as follows. First, the receiver measures the loss event rate (by using a losses history structure to record traces of received and lost packets) and sends it back to the sending entity. Then, the sender using this feedback measures the round-trip time. The sender also determines the sending rate using the TCP throughput Reno equation and varies its transmit rate to match this value. An enhancement of $TFRC$ rate control can be considered guided by QoS-aware consideration (TD-$TFRC$ Time Constraint and Differentiated Rate Control service). The aim is to provide the application with a suitable rate control policy consistent with both the delay constraints of the application and the end-to-end delay.

From these mechanisms one may consider several situations that should be considered for self-adapting strategies:

- s1: The slow start phase during which the rate increases exponentially
- s2: The congestion avoidance with a linear increasing of the rate
- s3: The detection of lost or explicit congestion notification (ECN) packets.
- s4: An abnormal value of the delay i.e. higher than a tolerated value during any of the previous situations
- s5: A maintained value of the delay higher than a tolerated value

The recognition of these different situations is a guide for the choice of the composition or recomposition of micro-protocols involved in the reconfiguration.

Situation recognition

A chronicle model is designed for each identified situation based on the chronicle formalism. A chronicle is defined by a conjunction of statements which are the declaration of event occurrences, time constraints between time points of the chronicle, declaration of guards, constraints on the variables of the chronicle and actions seen as the generation of specific events when the chronicle is recognized. Figure 4 gives an example of a chronicle definition.

A chronicle can also be defined by using messages. Messages define the kind of data that could be embedded in events and allow one to match a specified event in the input event flow. For instance $event(packet\_message, t)$ is an event based on a message named $packet\_message$ which should be received at the date t. A message may also include parameters and constraints. The message $packet\_message[?num, ?size]$ is then a message where the parameters are variables that should be instantiated by the chronicle recognition system when an event occurs. This is indicated by the symbol ?. Finally, an event can be defined based on such a message: $event(packet\_message[?num, ?size], t)$. In this case, the event is based on a message which should be received at date t and matches the message pattern.

In the remainder we present the chronicles associated to the situations $s3$ and $s5$.

For situation 3 *detection of loss or marked packets* $TFRC$ assumes that all packets contain a sequence number that is incremented by one for each packet that is sent and that a time stamp indicates when the packet is sent. If a lost packet is retransmitted, the retransmitted packet is given a new sequence number that is the latest in the transmission sequence. The loss of a packet is detected by the arrival of at least three packets with a higher sequence number than the lost packet. The corresponding chronicle is given figure 5. This chronicle generates an event ($packet\_loss$) when the detection is performed. num_loss is the sequence number of the lost packet. A mechanism is necessary to capture this data from the transport protocol. We assume that for each packet a message $packet[?num\_loss, ?seqn]$ is generated and gives the sequence number of the lost packet but also the sequence number of the packet itself. An event associated to the message packet must occur three times between $t1$ and $t2$ with the additional constraint that its sequence number ($seqn$) must be greater than the sequence number of the lost packet (num_loss). We use a counting predicate $occurs((n1, n2), M, (t1, t2))$ to describe that an event M must occur exactly N times between $t1$ and $t2$ with $n1 \leq N \leq n2$. $t1$, $t2$ are time symbol instantiated by the chronicle recognition system. Note that the chronicle could also be parameterized by a counter of occurrences instead of considering an *a priori* number of occurrences. In this way, the activation of a particular protocol mechanism guided by situation recognition could be performed based on external parameters such as the user requirements.

In the case of the situation $s5$, the chronicle generates an event when the delay is higher than the tol-

```
Message delay [?value ,?threshold]
    {
     }
Chronicle Tolerance  [ ?agreement,?threshold]
    {
      event (delay ( [?value,?threshold], t1)
      ?value>=?threshold
      when recognized {
            emit event(?agreement,t1);
          }
    }
```

Figure 6: Chronicle for the delay value

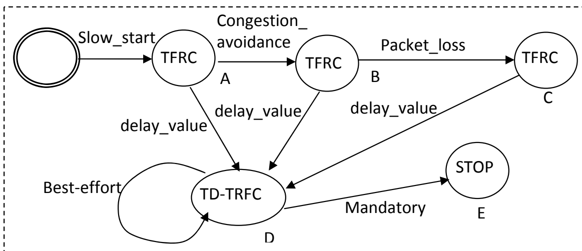

Figure 7: Intra-connection self-adapting strategy

erated value. The generated event by the chronicle is used for self_adapting strategy and then depends on the QoS transport protocol agreement: mandatory or best-effort (see figure 6).

The self_adapting strategy for intra-connection reconfiguration is guided by the situation recognition. It can be defined by a finite state machine (see figure 7) where the states are either the micro-protocol states (i.e. A, B and C for $TFRC$) or the micro-protocol itself (i.e. D for $TD-TFRC$). The evolutions between states are induced by the events generated by the chronicles i.e. by the diagnosis.

In this case study, the use of chronicles in the context of the TFRC micro-protocol is trivial because it illustrates the elementary behavior of a congestion controlled transport connection. The selection of micro-protocols allowing self-adaption is an open problem which depends on the complexity of the application and network contexts. In this example, the selection of the TD-TFRC micro-protocol based on the time-constrains expressed by the application requirements illustrates the benefits of using our approach to implement self-adaptation strategies. The next case study presents a more elaborated self-adaptation strategy.

4.2 Case study 2: Interconnection strategies

Interconnection strategies deal with collaborative management of multiple connections. This kind of strategies could be used for applications such as multimedia applications where several media streams compete for the network resources. In this case, the relevant situations to recognize are similar to the case study 1 for each connection, but the performed adaptation may concern only one connection or may involve several connections (sharing a common pool of limited resources). Let us suppose only two connections cx and cy.

- s1,s2,s3: Same as case study 1 for each connection
- s4x: For cx an abnormal value of the delay i.e. higher than a tolerated value during any of the previous situations
- s4y: For cy an abnormal value of the delay i.e. higher than a tolerated value during any of the previous situations
- s5x: At the level of cx, a maintained value of the delay higher than a tolerated value
- s5y: At the level of cy, a maintained value of the delay higher than a tolerated value

The self_adapting strategy for interconnection reconfiguration is based on the intra-connection reconfiguration strategy (from $TFRC$ to $TD-TFRC$) but also on load balancing between n intra-connections to switch from $TFRC$ to collaborative $TFRC$.

Indeed if at the level of the connection cx, a maintained value of the delay higher than a tolerated value is detected several strategies can be applied: TD-$TFRC$ on connection cx, load-balancing to use the other connection cy or both TD-$TFRC$ on cx and load-balancing on cy. For instance, in the case of a multimedia conferencing application, higher priority could be given to audio streams over video streams. In situations where the measured delay for the audio data delivery is too high to be accepted, video stream transmission could be degraded (e.g. by selectively discarding several video pictures) in order to privilege the audio stream. Moreover, audio data could be transmitted using both audio and video transport connections in order to offer a better service to the final user.

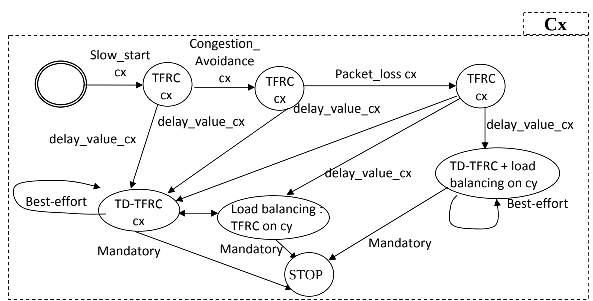

Figure 8: Interconnection self-adapting strategy: cx point of view

Figure 8 shows the finite state machine corresponding to the self-adapting strategy from the point of view of one connection cx, having a delay value greater than the required one. The intra-connection self-adapting strategy is still possible but an inter-connection strategy based on the load-balancing concept can be executed taking into account the collaborative capabilities between the connections. Here, the connection cx is going to benefit, for instance, of a rate reduction of the connection cy, the two connection sharing common overloaded resources (e.g. intermediate routers).

Let us note that the finite state machine illustrated on Figure 8 does not represent the behavior of cy.

The representation of the self-adapting strategy from a global point of view on all the connections is not trivial and several aspects need further investigation. The nondeterministic character of the state machine or the consideration of each connection priority are some examples of the problems to be addressed.

5 CONCLUSION

In this paper we have investigated the idea of applying a model-based diagnosis technique for network supervision and management. We elaborated chronicle-based models to represent the changes in the different traffic context situations that need to be detected for run-time adaptation of the communication protocols. The tractability of the approach is illustrated by modeling one of the event patterns encoded in the $TFRC$ protocol for the so-called intra-connection self-adapting strategy. The case of multiple connections and their cooperative management leading to interconnection strategies is also discussed and foreseen to be affordable as well. The idea of using parameterized chronicles to adapt to different degrees of requirement is proposed as the next issue to be investigated.

6 FUTURE WORK

The preliminary ideas presented in this paper must be further investigated and the relevance and efficiency of such an approach must be compared to other methods leading the composition of micro-protocols. Moreover, studies aimed at evaluating the scalability and the performance offered by this approach needs to be carried out.

The problem of building the chronicles must be considered. A way to go is to learn the chronicles. But chronicles can also be deduced from the mechanisms embedded in the standard monolithic protocols. One advantage of representing these mechanisms in the form of chronicles is to extract them from the protocols themselves. Another advantage is that chronicles may provide more flexibility. For instance, standard detection event patterns generally include constants provided by some counters. One may count for example the number of times that a packet is received with a sequence number superior to the expected one. These constants are fixed *a priori* and express an "average" loss along some QoS dimension that is desirable to be detected to trigger a protocol adaptation strategy.

Our ultimate goal is driven by QoS requirements stated on individual bases, also referred as QoE (Quality of Experience). The idea that is pursued is to be able to capture different users unsatisfaction profiles and to investigate the use of *parameterized chronicles*, i.e. chronicles in which the parameters can be adapted. Learning mechanisms can be implemented in order to estimate the parameter values for the different profiles. A degree of requirement could then be proposed to the users.

From a more general point of view, the approached problems open new perspectives for cooperative diagnosis, considering an evolutive system with dynamic observations, and in which faults may have an impact at different levels of the network and on the supported applications. In this context, QoS-driven diagnosis is a new topic that deserves particular attention.

The challenge of adapting the communication protocols for coping with dynamic situations is also an open field of investigation for diagnosability analysis as it introduces the need to question monitoring data integrity, which is not usual. In this sense it would be interesting to consider the design and implementation of signaling protocols required by reconfiguration policies in a way guided by diagnosis and adaptation.

Finally, the mobility aspect and then the topology of the networks should also be investigated to address the problem of self-adapting strategies in collaborative communicating systems.

ACKNOWLEDGMENTS

We would want to thank Christophe Dousson for his help to use the Chronicle's Language.

REFERENCES

(Allen, 1983) James F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832 – 843, 1983.

(Allen, 1984) James F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

(Bertrand *et al.*, 2008) O. Bertrand, P. Carle, and C. Choppy. Towards a coloured petri nets semantics of a chronicle language for distributed simulation processing. In *In CHINA 2008 Workshop (Concurrency metHods: Issues aNd Applications)*, pages 105–119, 2008.

(Carle *et al.*, 1998) P. Carle, P. Benhamou, M. Ornato, and FX. Dolbeau. Building dynamic organizations using intentions recognition. Technical Report ONERA-TP - 99-136, Office national d'études et de recherches aérospatiales, Châtillon, France, 1998.

(Carrault *et al.*, 1999) G. Carrault, M.-O. Cordier, R. Quiniou, M. Garreau, J.-J. Bellanger, and A. Bardou. A model-based approach for learning to identify cardiac arrhythmias. In G. Lindberg S. Andreassen W. Horn, Y. Shahar and J. Wyatt, editors, *Proceedings of AIMDM-99 : Artificial Intelligence in Medicine and Medical Decision Making*, 1999.

(Cordier and Dousson, 2000) MO Cordier and C Dousson. Alarm driven monitoring based on chronicles. In *4th Sumposium on Fault Detection Supervision and Safety for Technical Processes (SafeProcess)*, pages 286–291, Budapest, Hungary, june 2000.

(Cordier *et al.*, 2007) M.-O. Cordier, X. Le Guillou, S. Robin, L. Rozé, and T. Vidal. Distributed chronicles for on-line diagnosis of web services. In G. Biswas, X. Koutsoukos, and S. Abdelwahed, editors, *18th International Workshop on Principles of Diagnosis*, pages 37–44, May 2007.

(Dousson and Duong, 1999) C. Dousson and T. Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *IJCAI 99: Proceedings of the Sixteenth*

International Joint Conference on Artificial Intelligence, pages 620–626, San Francisco, CA, USA, June 1999.

(Dousson *et al.*, 1993) C. Dousson, P. Gaborit, and M. Ghallab. In *IJCAI: International Joint Conference on Artificial Intelligence*, pages 166–172, Chambéry, France, august 1993.

(Guerraz and Dousson, 2004) B. Guerraz and C. Dousson. Chronicles construction starting from the fault model of the system to diagnose. In *International Workshop on Principles of Diagnosis (DX04)*, pages 51–56, Carcassonne, France, 2004.

(Kowalski and Sergot, 1986) Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.

(Laborie and Krivine, 1997) P. Laborie and J.-P. Krivine. Automatic generation of chronicles and its application to alarm processing in power distribution systems. In *8th international workshop of diagnosis (DX97)*, Mont Saint-Michel, France, 1997.

(Mayer, 1998) E. Mayer. Inductive learning of chronicles. In *European Conference on Artificial Intelligence*, pages 471–472, 1998.

(McCarthy and Hayes, 1969) J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence*, 4, 1969.

(Morin and Debar, 2003) Benjamin Morin and Hervé Debar. Correltaion on intrusion: an application od chronicles. In *6th International Conference on recent Advances in Intrusion Detection RAID*, Pittsburgh, USA, september 2003.

(Pencolé and Subias, 2009) Y. Pencolé and A. Subias. A chronicle-based diagnosability approach for discrete timed-event systems: Application to web-services. *Journal of Universal Computer Science*, 15(17):3246–3272, 2009.

(Rota and Thonnat, 2000) N. Rota and M. Thonnat. Activity recognition from video sequences using declarative models. In *14th International Workshop on Principles of Diagnosis (DX00)*, Morelia, Michoacen, Mexico, June 2000.

(VanWambeke *et al.*, 2007) N. VanWambeke, E. Exposito, and M. Diaz. Transport layer qos protocols: the micro-protocol approach. March 2007.

(VanWambeke *et al.*, 2008) N. VanWambeke, F. Armando, C. Chassot C, and E. Exposito. A model-based approach for self-adaptive transport protocols. *Elsevier Computer Communications, Special issue on end-to-end support over heterogeneous wired and wireless network*, 31(11):2699–2705, July 2008.

(Zimmermann, 1980) H. Zimmermann. Osi reference model: the iso model of architecture for open systems interconnection. *IEEE transactions on communication*, 28(4), 1980.