Smart highways sensor network modeling: Real-time sensor fault detection

Abdelaziz Daaif † , Omar Bouattane†, Mohamed Youssfi† and Sidi Mohamed Snineh†,

SSDIA Laboratory ENSET Mohammedia, Hassan II University of Casablanca, Morocco

Summary
Smart highways use a variety of sensors to capture the passage of vehicles and send events to the remote processing center in real time. The failure of a sensor can lead to processing errors and unfortunate decisions. This paper proposes a fault-tolerant architectural model, which facilitates the reasoning on large networks of highway sensors. This model is based on the aggregation of a set of physical sensors into a single logical sensor. The proposed approach allows to map a logical sensor to each node of the highways network graph. It also allows to implement real-time fault detection and correction procedures at the level of the logical sensors. To validate the proposed approach, some simulation results of a failure detection in the "input / output" component of a highway are presented to show the effectiveness of this model.

Key words:
Realtime event processing; Smart highways; Distributed processing; Sensors network; Sensors fault detection.

1. Introduction

Nowadays, surveillance, control and management of traffic in highways has become vital [1][2][3]. Mismanagement leads to considerable losses in travel time and energy and increases the risk of accidents and air pollution [4]. In recent years, there has been considerable development in the field of road traffic sensors. In [5][6] the authors present a road traffic sensor that integrates advanced embedded components, managed by distinctive algorithms for the implementation of various traffic monitoring applications. The sensor can detect, count, accurately estimate speed and length, and classify vehicles in real time. In [7], authors present an instrumentation model of smart highways by breaking down highways into hierarchical cells. However, their approach is based on the determination of sensor locations and does not show the instrumentation of the various components of a highway network.

Highways are increasingly instrumented and generate huge data from a multitude of sensors. Several fusion techniques use multi-source data to estimate the state of the traffic. In [8], the authors combine the location data of the users' mobile phones and the data captured by the loop detectors to accurately assess the state of the road. In [9], authors mixed data coming from connected vehicles and on-site sensors to estimate traffic state. Most research studies assume that the sensors are perfect and do not consider in their models the case of failure of one or more sensors. The highway networks are modeled by oriented graphs and the events generated by the sensors are associated with one or more nodes of the graph. When a sensor fails, the traffic state computational models must detect the failure and adapt the graph to the new situation.

The present work is closely based on the highway networks model used in [10]. Thus, we propose an architectural approach that maps each node of the highway network graph to a logical sensor. In this model, a logical sensor is a processing unit associated with a set of physical sensors. It is equipped with a processing capacity to manage all of its components and a communication channel connecting it to the remote processing center. This approach reveals several advantages as:

- It facilitates the reasoning on the global topology of the logical sensors, since this topology corresponds exactly to the graph of the highways network,
- The synchronization of the events is done only at the logical sensors level,
- The detection of the physical sensors failures is done at the logical sensor level,
- It reduces the communication costs with the remote processing center by collecting multiple events in a single message,
- The logical sensors can implement algorithms for vehicle counting, classification and keep a history on which they can relay in case of failure of one or several sensors since they are equipped with storage, processing and communication capabilities.

In the following section, we will define the components of a highways network and the model of composition of their logical sensors. Section 3 is dedicated to the study of failure cases and the approach used for its detection. The last section gives some concluding remarks.

2. Architectural model of smart highways

2.1 Highways network components

A highway is described by a sequence of components that always starts with an entry, followed by several intermediate nodes, and ends with an exit. Intermediate

Manuscript received December 5, 2017
Manuscript revised December 20, 2017

nodes may be entrance / exit, service areas, tolls or exchangers. The exchangers make it possible to interconnect two or more highways. See Figure 1. All these components are instrumented by vehicle passage sensors. To increase the accuracy of traffic monitoring processes, sensors are interposed between the components according to the distances separating them and the intended accuracy.

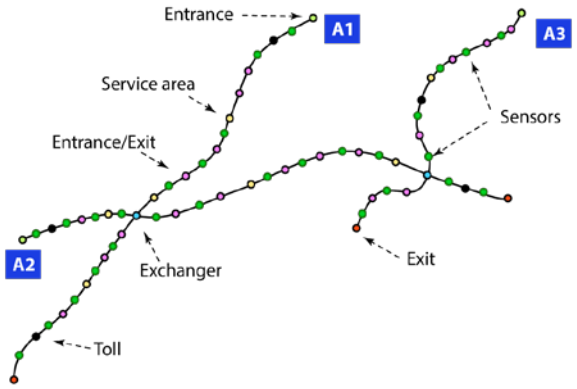


Fig. 1 Smart highways components

Components and sensors are modeled as logical sensors. Each sensor has a unique identifier. The highways network is modeled by an oriented graph whose vertices are represented by logical sensors and edges by highway segments. See Figure 2.

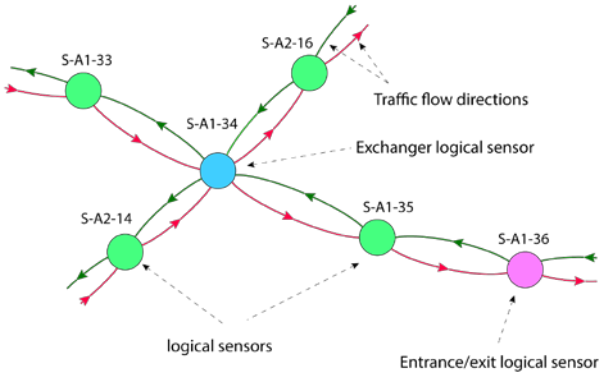


Fig. 2 Highways sensors graph

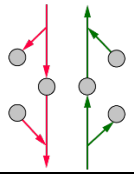
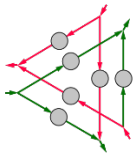
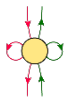
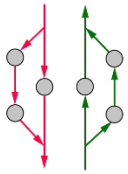
2.2 Logical sensor model

The highway network is broken down into a set of segments that are interconnected by sensors. The goal is to count in real time the number of vehicles in each segment using the conservation law. The composition of a logical sensor is realized so that a physical sensor must not connect more than two segments.

To be able to identify a vehicle passing, the sensor must generate an event that uniquely identifies the direction taken by the vehicle. As can be seen in Figure 2, a vehicle traveling from S-A1-35 sensor to SA-1-33 must pass through the S-A1-34 sensor. The latter will generate an event that will be identified by the triplet (S-A1-35, S-A1-34, S-A1-33). More generally, the passage of a vehicle through a logical sensor having the sidi identifier generates an event identified by (sidi-1, sidi, sidi+1).

For each component type of a highway, the logical sensor must be decomposable into a set of physical sensors to distinguish the different possible paths. See Table I .

Table 1: Physical sensors

Component	Logical sensor	Physical sensors
Entrance		
Entrance/Exit		
Exchanger		
Srvice Area		
Sensor / Toll		
Exit		

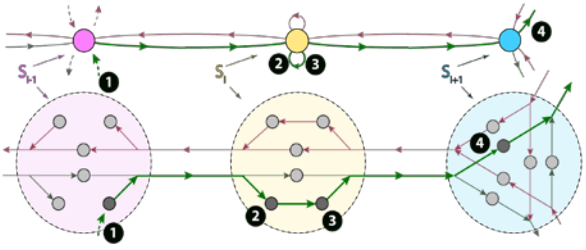


Fig. 3 Generated events during a given route

Each event generated by a physical sensor contains:

- The event identifier (sid_{i-1}, sid_i, sid_{i+1})
- The speed of vehicle

- The passage date (Timestamp)
- Other information depending on the sensor characteristics.

Figure 3 shows the example of a route starting with an Entrance / Exit (step 1 - logical sensor Si-1), stopping in a service area (steps 2 and 3 - logical sensor Si) and then going through an exchanger (step 4 - logical sensor Si+1). Table 2 presents for each step, the event identifier.

Table 2: Event identifier

Step	Logical sensor	Event identifier
1	Si-1	(Null, S_{i-1}, S_i)
2	S_i	(S_{i-1}, S_i, S_i)
3	S_i	(S_i, S_i, S_{i+1})
4	S_{i+1}	(S_i, S_{i+1}, S_{i+2})

For logical sensor architecture, each logical sensor type consists of a set of physical sensors and a communication unit (CU) that sends the generated events to the processing center (PC). See Figure 3. To facilitate the synchronization and implementation of the sensors, the events are timestamped in the CU.

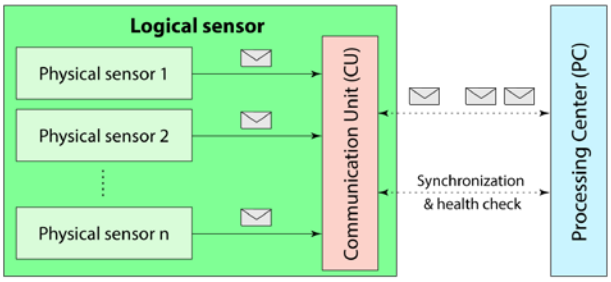


Fig. 4 Logical sensor architecture

3. Detection of faulty sensors

3.1 Failure scenarios

Failures can occur at multiple levels of distributed sensor architecture. We list below the various possible fault locations:

1. Failure of communication between CU and PC,
2. Failure of CU,
3. Failure of some physical sensors or their communication link with the CU.

The first two cases represent the failure of the entire logical sensor (global failure). The third case represents the failure of some of the physical sensors (partial failure).
During an overall sensor failure, the entire sensor is disabled and the graph is updated to reflect the new situation. The adjacent logical sensors to the defective sensor are not aware of this fact, they continue to generate events referring to this sensor.

At any time, a logical sensor can only be in one of these three states (Figure 5):

- Functional (**FS**)
- Global failure (**GF**)
- Partial failure (**PF**)

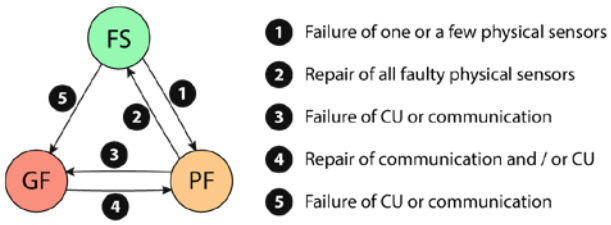


Fig.5 The three states of logical sensors

Transitions 1 and 2 of Figure 5 are detected locally by the sensor itself while transitions 3, 4 and 5 are detected by a remote procedure that runs in the processing center.

3.2 Failure detection

3.2.1 Local detection

For each type of logic sensor, we use the conservation law to determine the defective physical sensors. See Eq. 1 and Figure 6.

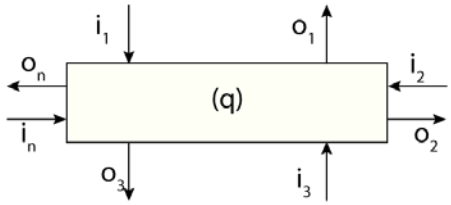


Fig. 6. Conservation law

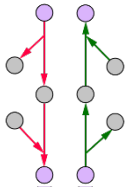
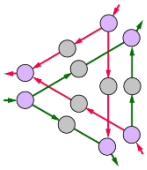
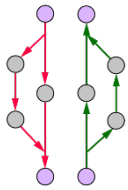
$$\Sigma\, o_j - \Sigma\, i_j = q \qquad (1)$$

With:

q: number of vehicles inside the sensor
o_j: number of vehicles left
i_j: number of vehicles entered

To be able to use the conservation law, additional physical sensors must be added to each type of logical sensor. Table 3 summarizes the modifications made to the different logical sensors.

Table 3: Modifications made to logical sensors

Component	Logical sensor	Physical sensors
Entrance		
Entrance/ Exit		
Exchanger		
Service area		
Sensor		
Exit		

The addition of further sensors makes it possible to implement failure detection routines, but also makes the component more efficient and fault-tolerant.

UC uses a streaming-oriented approach (See Figure 7). When an event arrives from a physical sensor, it is first timestamped and then used on the one hand to reset the heartbeat timer, on the other hand to feed the failure detection procedure (FDP). The latter performs its procedure and in case of failure, requests a health check at the faulty sensor. The Event Filtering and Correction (EFC) procedure filters events according to the state of the physical sensors. In the case of a failure, this procedure attempts to correct the events based on the sensors state and counter values provided by FDP. Finally, the selected event is enriched with status information and sent to the PC.

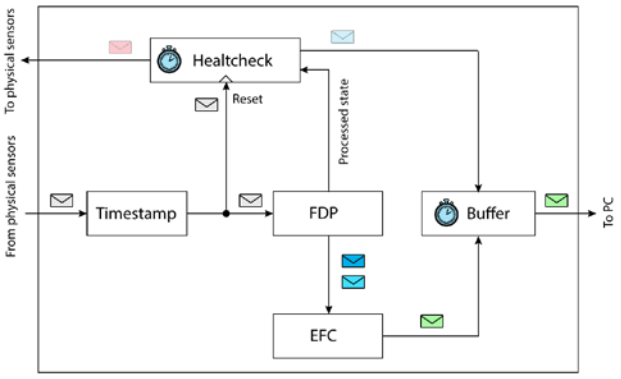


Fig. 7 Communication unit (CU)

In what follows, we propose the detection approach used for the detection of failures in the logical sensors "Entrance" and "Entrance / Exit". The other sensors use the same approach.

a) Logical sensor: Entrance

We assume that the distance between the physical sensors in each direction is negligible (See Figure 8). The number of vehicles entering is equal to the number of vehicles leaving ($i_1 = o_1$ and $i_2 = o_2$). For each direction, we associate a variable (respectively d_1 and d_2). When a vehicle enters, we increment the corresponding variable and decrement it when it comes out.

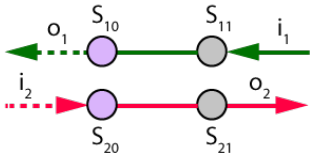


Fig. 8 Logical sensor « Entrance »

If all the sensors are working correctly, both variables must have one of the values (-1, 0 or 1). For any other value, the sign of the variable determines which of the two sensors is defective.

In the real case, the distance between the physical sensors is not null and at the time of putting into operation of the detection procedure, it is not known the number of vehicles present between the two sensors ($d_1 = i_1 - o_1 = q_1$ and $d_2 = i_2 - o_2 = q_2$). The quantities q_1 and q_2 will fluctuate and remain within the range $[-q_{max}, q_{max}]$. The value q_{max} represents the maximum number of vehicles that can be intercalated between the two sensors.

b) Logical sensor: Entrance / Exit

The same reasoning applies to both directions. The conservation law is used for “direction 1”:

$$\begin{cases} i_{11} - o_{12} - m_1 = q_{11} & (2.1) \\ m_1 + i_{12} - o_{11} = q_{12} & (2.2) \\ i_{11} + i_{12} - o_{11} - o_{12} = q_1 = q_{11} + q_{12} & (2.3) \end{cases}$$

q_1, q_{11} and q_{12} represent the number of vehicles respectively in the meshes (i_{11}, o_{12}, m_1) (Eq. 2.1), (i_{11}, o_{12}, m_1) (Eq. 2.2) and (i_{11}, i_{12}, o_{12}, o_{11}) (Eq. 2.3).

A sensor s_i is either functional and noted S_i or faulty and noted $\neg S_i$.

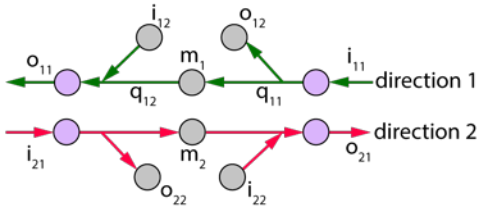


Fig. 9 Logical sensor "Entrance / Exit"

If we associate variables to the quantities q_1, q_{11} and q_{12} by incrementing them in the case of an input and decrementing them in the case of an output, these three variables should fluctuate in a range depending on the distances between the sensors. In this case, the variable is said stable. If at least one of these values diverges continuously, this means that one or more sensors are failing. In this case the variable will become unstable and will diverge to $-\infty$ or $+\infty$.

As long as there is vehicular traffic in the node and the variables q_1, q_{11} and q_{12} remain stable, we can say with certainty that all physical sensors are operational. But if some or all of these variables become unstable, then one or more sensors fail. We assume that physical sensors can perform a health test and say whether they are operational or not. If there is no traffic for a long time or when the logic sensor detects a failure, the sensor requests the physical sensors to perform this test to inform the remote center of the state of the sensors.

In the situation where a single sensor breaks down, we detect it without having recourse to the health test. In the following, we present the case where a single sensor breaks down.

Variables q_1, q_{11} and q_{12} can be in one of three states: stable, unstable positive or unstable negative. We represent these three states with a two-bit binary variable q_{b1}, q_{b11} and q_{b12}. (unstable negative: **00**, unstable positive: **01** and stable: **10** and **11**. The most significant bit expresses stability **1**X = stable, **0**X = unstable, where X=0/1. When the variable is unstable, the least significant bit expresses the type of instability; positive 0**1** or negative 0**0**)

Several cases are possible, Table 4 gives a summary of these cases (x means not applicable).

Table 4: Failure cases

	00	**1X**	**01**
q_{b11}	$\neg I_{11}$	I_{11}, O_{12}, M_1	$\neg O_{12}$ or $\neg M1$
q_{b12}	$\neg M_1$ or $\neg I_{12}$	M_1, I_{12}, I_{11}	$\neg O_{11}$
q_{b1}	See **Table 5**	See **Table 6**	See **Table 7**

Table 5: Case where q1 is unstable negative

q_{b1}=00		q_{b12}		
		00	**1X**	**01**
q_{b11}	**00**	$\neg I_{11}$ or $\neg I_{12}$	$\neg I_{11}$	x
	1X	$\neg I_{12}$	x	x
	01	x	x	x

Table 6: Case where q1 is stable

q_{b1} = 1X		q_{b12}		
		00	**1X**	**01**
q_{b11}	**00**	x	x	$\neg M_1$
	1X	x	ok	x
	01	x	x	x

Table 7: Case where q1 is unstable positive

q_{b1} = 01		q_{b12}		
		00	**1X**	**01**
q_{b11}	**00**	x	x	x
	1X	x	x	$\neg O_{11}$
	01	x	$\neg O_{12}$	$\neg O_{11}$ or $\neg O_{12}$

Table 8: Physical sensors failure conditions

Sensors	**Failure conditions**
$\neg I_{11}$	(q_{b11} == 00)
$\neg I_{12}$	(q_{b12} == 00) and (q_{b11} == 1X)
$\neg O_{11}$	(q_{b12} == 01)
$\neg O_{12}$	(q_{b11} == 01) and (q_{b12} == 1X)
$\neg M_1$	(q_{b1} == 1X) and ((q_{b11} == 00) and (q_{b12} == 01)

Table 8 gives the failure conditions for each logical sensor. For each message received, the FDP calculates the new state of the variables q_{b1}, q_{b11} and q_{b12}, performs the tests and updates the sensor state vector. ("**==**" and "**and**" are logical operators)

We have tested the case of direction 1 of Figure 10. Vehicles arrive randomly at inputs I_{11} and I_{12} with random speeds between two limit values. From I_{11}, the vehicle decides to exit with a probability of 0.3 or to continue with a probability of 0.7. Depending on the physical sensor, we increment or decrement the variables q_1, q_{11} and q_{12}. At a given time, several vehicles pass through the logical sensor in a concurrent manner, which makes it necessary to synchronize these variables. We use Javascript to develop this test, since its engine uses a single event loop based on a single thread.

We simulated the consecutive failures of the sensors in the order {I_{11}, O_{12}, M_1, I_{12}, O_{11}}. All failures are separated by a stability period.

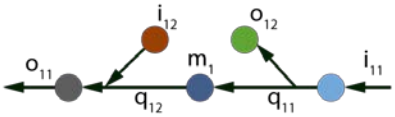


Fig. 10 Sensors in the direction 1 of the entrance/exit component

Figure 11 shows the passing case of 1840 vehicles generating 3343 events. The sensors I_{11}, O_{12}, M_1, I_{12} and

O_{11} are shown during their period of failure. When I_{11} breaks down, we can see that the variables q_1 and q_{12} diverge negatively and that the variable q_{12} remains stable. This is exactly to the given value of Table 5. The other cases of failure also satisfy the values given in Tables 5, 6 and 7.

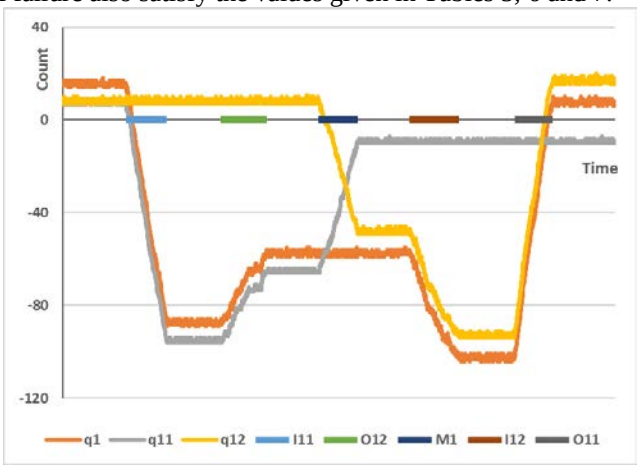


Fig. 11 Succession of failures of physical sensors without resetting variables q_1, q_{11} and q_{12}

Figure 12 shows the same simulation as that of Figure 11, with the difference that the variables are reset at the beginning of each stability period.

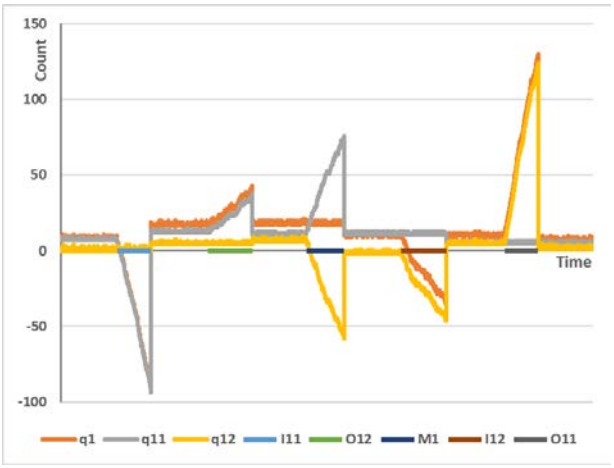


Fig. 12 Succession of physical sensor failures, with reset of variables q_1, q_{11} and q_{12}

In many cases of physical sensor failures, the logical sensor remains operational. Indeed, the EFC procedure (Figure 7) uses the captured values of variables q_1, q_{11} and q_{12} to derive the missing event. In the case of I_{11} sensor failure, the number of missed events can be determined from the relation $q_{11} = i_{11} - o_{12} - m_1$ (Eq.2.1).

If we consider that the distances between the sensors are null (the initial quantities of q_1, q_{11} and q_{12} are null). Variable q_{11} represents only the event count of the M_1 and O_{12} sensors. The missed value of i_{11} is $-q_{11}$.

Table 9 shows the recovery values in the event of a single sensor failure.

Table 9: Recovering missing values in the case of a single faulty sensor

Sensors	Failure conditions
$\neg I_{11}$	$i_{11} = -q_{11} = -q_1$
$\neg I_{12}$	$i_{12} = -q_{12} = -q_1$
$\neg O_{11}$	$o_{11} = q_{12} = q_1$
$\neg O_{12}$	$o_{12} = q_{11} = q_1$
$\neg M_1$	$M_1 = q_{11} = -q_{12}$

In case of failure of two sensors, it is always possible to recover the missing values of the sensors. Figure 13 shows the simultaneous failure cases of two sensors. Table 10 summarizes these cases and shows how to recover the missed values. (X values not recovered)

Table 10: Recovery of missing values in the case of two faulty sensors

	$\neg I_{12}$	$\neg O_{11}$	$\neg O_{12}$	$\neg M_1$
$\neg I_{11}$	$i_{11} = -q_{11}$ $i_{12} = -q_{12}$	$i_{11} = -q_{11}$ $o_{11} = q_{12}$	X	$m_1 = -q_{12}$ $i_{11} = -q_1$
$\neg I_{12}$		X	$i_{12} = -q_{12}$ $o_{11} = q_{12}$	$m_1 = q_{11}$ $i_{12} = -q_1$
$\neg O_{11}$			$o_{11} = q_{12}$ $o_{12} = q_{11}$	$m_1 = q_{11}$ $o_{11} = q_1$
$\neg O_{12}$				$m_1 = -q_{12}$ $o_{12} = q_1$

In case of failure of the pairs (I_{11}, O_{12}) and (I_{12}, O_{11}), the missed values are not recoverable. In these cases, either we consider that the sensor fails globally, or we rely on the historical data of the sensor to estimate the missing values.

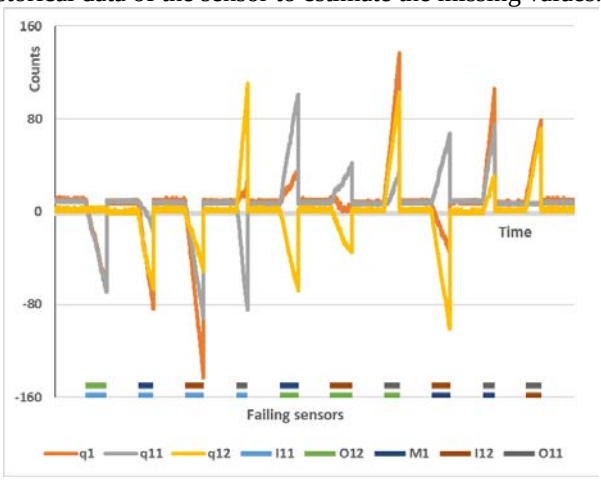


Fig. 13 Dual failure of sensors

3.2.2 Remote detection

The remote processing center receives the messages and ingests them into the streaming platform. See Figure 14. The detection procedure retrieves a copy of each message and checks its status. If it detects a partial failure, it reports it by generating an alert message and updates the highway network graph. Depending on the changes made to the graph, it may be necessary to correct the events before pushing them into the traffic estimation topology in the network segments.

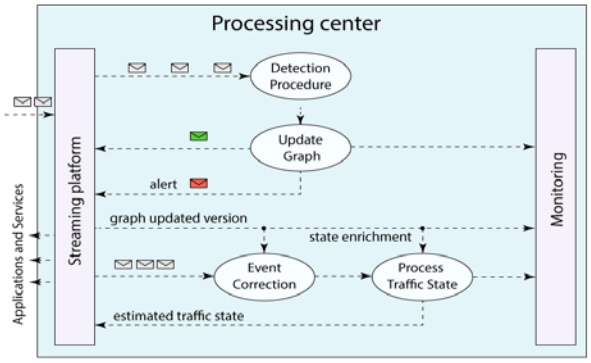


Fig. 14 Remote processing center architecture

When a logical sensor does not detect a vehicle for a given time period (Figure 7), the CU sends a heartbeat message containing the last state of the physical sensors. To detect global sensor failures, the detection procedure in Figure 14 records the last date of the last sensor message in each node of the graph. Since each event is identified by the triplet containing the predecessor sensor, the current sensor and the next sensor. The dates of the last message coming from the predecessor and the successor are compared to the current date. If this comparison exceeds a certain threshold, it means that the sensor has failed. In the case of detection of a fault, the adjacent sensor is also checked and so on.

4. Conclusion

Highway networks are increasingly instrumented by adding different types of sensors at different locations. This generates a flow of huge and disparate events that must be processed in real time to provide an estimate of the state of the highways. In the case of a sensor failure, calculations can lead to erroneous decisions. In this article, we have proposed an approach that allows to map a node of the highway network graph to a logical sensor. Depending on the highway components, abstractions of logical sensors are defined using the composition. Each logical sensor has a processing and communication capability that allows it to manage its physical constituents and perform fault detection routines. Using the conservation law, we have determined the failure conditions of physical sensors and we have shown that the use of a streaming-oriented approach is appropriate in this case. We believe that adopting this approach will facilitate management and reasoning in major highway networks. We plan in future works to propose a detailed model of the remote processing platform by showing the incidences of failures on the graph of the highway network.

References

[1] M. Pucher et al., "Multimodal highway monitoring for robust incident detection," 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, 2010, pp. 837-842.

[2] F. J. Villanueva, J. Albusac, L. Jiménez, D. Villa and J. C. López, "Architecture for Smart Highway Real Time Monitoring," 2013 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona, 2013, pp. 1277-1282.

[3] M. Fountoulakis, N. Bekiaris-Liberis, C. Roncoli, I. Papamichail and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 1761-1766.

[4] Csikós, Alfréd & Varga, István. (2012). Real-time Modeling and Control Objective Analysis of Motorway Emissions. Procedia - Social and Behavioral Sciences. 54. 1027-1036. 10.1016/j.sbspro.2012.09.818. doi: 10.1109/ITSC.2015.265

[5] W. Balid, H. Tafish and H. H. Refai, "Development of Portable Wireless Sensor Network System for Real-Time Traffic Surveillance," 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, 2015, pp. 1630-1637.

[6] W. Balid, H. Tafish and H. H. Refai, "Intelligent Vehicle Counting and Classification Sensor for Real-Time Traffic Surveillance," in IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1-11. doi: 10.1109/TITS.2017.2741507

[7] S. Ghosh, S. Rao and B. Venkiteswaran, "Sensor Network Design for Smart Highways," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 42, no. 5, pp. 1291-1300, Sept. 2012. doi: 10.1109/TSMCA.2012.2187185

[8] B. Jin, Y. Cui and F. Zhang, "Fusing Static and Roving Sensor Data for Detecting Highway Traffic Conditions in Real Time," 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, 2016, pp. 807-816. doi: 10.1109/COMPSAC.2016.120

[9] M. Fountoulakis, N. Bekiaris-Liberis, C. Roncoli, I. Papamichail and M. Papageorgiou, "Highway traffic state estimation with mixed connected and conventional vehicles: Microscopic simulation-based testing," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 1761-1766. doi: 10.1109/ITSC.2016.7795796

[10] Abdelaziz Daaif, Omar Bouattane, Mohamed Youssfi and Oum El Kheir Abra, “An Efficient Distributed Traffic Events Generator for Smart Highways” International Journal of Advanced Computer Science and Applications(ijacsa), 8(7), 2017. http://dx.doi.org/10.14569/IJACSA.2017.080717

Abdelaziz Daaif was born in 1965 in Ouaouizerth, Morocco. He is an associate professor in the Department of Mathematics and Computer Science, ENSET, Hassan II University in Casablanca since 1994. He received his master's degree in "Distributed information systems." in 2014. Currently, he is a Phd student. His research focuses on parallel and distributed systems, multi-agent systems, Semantic Web, High Performance Computing and Software Engineering.

Omar Bouattane received his PhD from the University Hassan II of Casablanca, Morocco in 2001 in Parallel Computing and Image processing. He currently serves as a full Professor in the Department of Electrical Engineering at ENSET of Mohammedia. He has more than 100 scientific publications in various domains of Computational Intelligence, high performance computing, image processing parallel computing and renewable energy. He has registered 6 national and PCT international Patents regarding to the networking technology and signal synthesis. He was awarded as the owner of the best PCT patent in Morocco on 2011. Since 2012, He was the head of the laboratory of Signals Distributed Systems and Artificial Intelligence. He involved his laboratory in several partnership activities and developed many funded projects in Morocco and in his university.

Mohamed Youssfi was born in 1970 at Ouarzazate, in Morocco. He serves as Professor Researcher in Department of Mathematics and Computer Science, ENSET, Hassan II University of Casablanca since 1996. He received his PhD (Doctorat de 3ème cycle) from The Mohammed V University of Rabat, Morocco, 1996 in Computer Science and his PhD (Doctorat d'Etat) from The Mohammed V University of Rabat, Morocco, 2015 in Parallel and Distributed Systems. His research is focused on various domains of Computational Intelligence as Parallel and Distributed Systems, Parallel Architectures, Multi Agent Systems, Web Semantic, High Performance Computing and Software Engineering.

Sidi Mohamed Snineh was born in 1968 in Casablanca, Morocco. He is a professor in the Department of Mathematics and Computer Science, CRMEF in Marrakech since 1998. He obtained two masters, one in "Engineering and Management of Information Systems" from the University Toulouse I Social Sciences in 2006 and the other in "Distributed Information Systems" in 2016. Currently, he is a PhD student in the second year. His research focuses on parallel and distributed systems, multi-agent systems, semantic Web, high performance computing.