AIAA Guidance, Navigation, and Control Conference
2 - 5 August 2010, Toronto, Ontario Canada

AIAA 2010-8329

Cooperative Communication Failure Detection for Multiple Vehicle Systems

Hojjat A. Izadi[1], Brandon W. Gordon[2] and Youmin Zhang[3]
Concordia University, Montreal, Quebec, H3G 1M8, Canada

In this paper, new cooperative fault detection algorithms are developed to detect the inter-vehicle communication failures in a network of multiple vehicle systems. The proposed detection algorithms work based on the notion that communication failure can lead to breaks or delays in the exchanged messages. Depending on the communication topology and the communication devices employed, different algorithms are required to diagnose failure and identify the faulty vehicles in the group. Algorithms are developed for the cases with communication that is bidirectional, unidirectional, and employed with separate transmitter and receiver units. For each algorithm, the necessary conditions on the communication graph topology, under which failure is detectable, are derived. Using probability analysis the reliability of the proposed algorithms is also investigated.

Nomenclature

d	=	discrete communication time delay
$\mathbb{E}$	=	set of edges in graph topology
$\mathbb{G}$	=	graph topology
N_v	=	number of vehicles
N_f^i	=	number of followers of vehicle *i*
N_l^i	=	number of leaders of vehicle *i*
N_n^i	=	number of neighbors of vehicle *i*
$\mathbb{V}$	=	set of vehicles
δ	=	sampling period
τ	=	communication delay
τ_l	=	large communication delay
τ_s	=	small communication delay

I. Introduction

Cooperative fault detection has recently become the interest of many research areas including the network of measurement sensors,[1-4] team of multiple unmanned vehicles,[5] network of computers,[6-8] cooperative robot manipulators,[9] and network of security cameras[10]. For such network of interacting subsystems, some degrees of cooperation for diagnosing the possible failures have shown superior security and reliability rather than using non-cooperative approaches. For example, in Ref. 4, a cooperative fault-detection mechanism is proposed for detecting failures in underwater sensor networks; in this framework each member of network independently detects the fault status of its sensor and then employs a distributed agreement protocol to reach an agreement on the fault status. Also, in Ref. 5 a cooperative sensor fault detection and identification method is presented for a team of Unmanned Aerial Vehicles (UAV). In order to increase the reliability of sensor fault detection and identification, the method benefits from the capabilities that the team of UAVs offers, by using the additional data from sensors of other UAVs. Furthermore, in Ref. 9 for cooperative robotic manipulators a cooperative fault

[1] Postdoctoral Fellow, Department of Mechanical and Industrial Engineering, 1515 St. Catherine West, ha_izadi@encs.concordia.ca.
[2] Associate Professor, Department of Mechanical and Industrial Engineering, 1515 St. Catherine West, bwgordon@encs.concordia.ca.
[3] Associate Professor, Department of Mechanical and Industrial Engineering, 1515 St. Catherine West, ymzhang@encs.concordia.ca, Senior Member AIAA.

detection framework is developed to detect the joint position and velocity sensors faults. The position and velocity sensor faults are detected by analyzing the position and velocity constraints in connected joints of manipulators. If the difference between the measurement of one sensor and other sensors are larger than some threshold a failure is concluded.

Recently, the practical implementation issues such as computation time, communication requirements, and model uncertainties in the field of cooperative vehicle system control have been considered widely.[11-17] Towards considering practical implementation issues, in this paper the communication fault detection over a network of cooperative vehicles operating in a decentralized fashion is considered. The communication failure can potentially lead to break/delay in the communicated messages. Some examples of the communication failures leading to large communication delays for the team of cooperative vehicles can be found in Refs. 11, 18-19. In Ref. 18, the wireless communication packet loss/delay is considered; also, in Ref. 11, the communication failure in the formation flight of multiple UAVs leads to break in the communicated messages that enforce the fleet to redefine the communication topology.

In this paper, it is assumed that each vehicle is equipped with a high performance communication device and the cooperative vehicles share a communication channel[22] to communicate to each other. It is also assumed that in the normal conditions the high performance communication device enables vehicles to communicate with neighboring vehicles with a very small delay, typically smaller than the sampling time. Then, the faulty condition is defined as follows:

Faulty condition: The high performance communication device of one vehicle in the team fails.

It is desired to provide some cooperation among the neighboring vehicles to diagnose the defined communication failure.

A few research works have addressed the communication failure detection for multiple vehicles. A very closely related work is presented in Ref. 11 where it is desired to manage the communication failures in formation flight of multiple UAVs; it is assumed that the communication failure leads to complete blockage of information flow to/from faulty vehicle. Then, in Ref. 11 to keep all aircrafts informed about all operational members in the group, it is suggested to use an extra broadcasting communication channel. If after some specific time one aircraft has not sent its "*alive*" signal through the backup communication channel, that aircraft is considered lost. In another related work,[18] two faults for formation flight of UAVs are considered: 1) GPS sensor failure and 2) wireless communication packet losses. To detect the GPS sensor failure a state/output observer is used which monitors the behavior of a UAV. The output of the observer is compared with the GPS data, and if the difference is larger than some threshold then a GPS fault is identified. Also, in Ref. 18 to detect the communication packet loss/delay, the faults are identified by numbering the packets sequentially and the number of the packet is also transmitted; a mismatch between the expected packet number and the received packet number implies the occurrence of a fault (packet loss).

The communication failure detection of cooperative vehicles becomes more challenging when a decentralized structure is used, as every vehicle should rely on the local information. Also, depending on the choice of communication topology (unidirectional or bidirectional) or whether the failure happens to transmitter or receiver devices, different scenarios can happen. In this paper three main cases are considered: 1) unidirectional communication (*directed* topology) 2) bidirectional communication (*undirected* communication topology) 3) separate *transmitter* and/or *receiver* failures (unidirectional or bidirectional). The latter considers the cases where the failure does not necessarily apply to both receiver and transmitter devices.

The failure diagnosis algorithm for each vehicle includes: 1) Monitoring and detecting the faulty situation, 2) Identifying the faulty vehicle in the team; *i.e.* each vehicle must determine if itself is faulty or its neighbors by cooperating with neighbors, it must also determine which neighbor is faulty.

II. Communication System

Every communication system has three main units[22]: *transmitter (TX)*, *receiver (RX)* and *channel*. Figure 1 shows a general schematic of a communication system. *Transmitter* is a device, installed on the source vehicle, which converts the messages to the suitable signals such as electromagnetic signals (in wireless communication, using an antenna the *transmitter* propagates this signal). A *receiver* device, which is installed on the destination vehicle, has an operation inverse to the operation of *transmitter*; radio is a typical receiver. A *channel* is a medium used to carry the information from *transmitter* to *receiver*[22]; it can be a bond of frequencies or light or anything which can carry the signal.

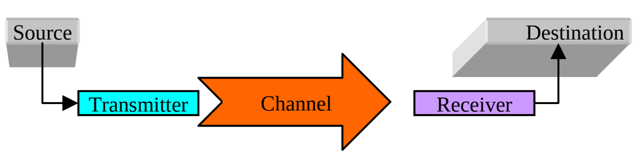

Figure 1. *Schematic of communication system*

It is assumed in this paper that each vehicle in the team receives/sends the information from/to neighbors. Hence, each vehicle needs both *transmitter* and *receiver* units. The communication system may have both these parties connected similar to *duplex communication systems*; or two separate devices may be used for TX and RX. In this paper, both cases are addressed.

More precisely, the communication failure is referred to as the failure or malfunction of the *transmitter-receiver* devices. If they are embedded in one single unit then the failure of each one implies the failure of both.

III. Communication Topologies using Graph Theory

In the cooperative control, there may exist different interactions between vehicles. As some examples of such interactions one can point to physical interactions such as collision avoidance and communication. The "*graph theory*" is a relevant mathematical tool for representing and modeling the interactions; the interaction between cooperative vehicles is usually represented by an "*interaction graph*" which is described by two basic elements: *nodes* and *edges*, where the *nodes* represent the subsystems/vehicles and an *edge* between two nodes denotes the interaction between these two subsystems. The "*interaction graph*" can be *directed* (unidirectional) or *undirected* (bidirectional).

Considering a set of N_v vehicles cooperating to perform a common mission, the i^{th} vehicle in the team is associated to the i^{th} node of the graph. If an edge (i, j) connecting the i^{th} and j^{th} node is present, it means the i^{th} and j^{th} vehicles have an interaction; this relation is termed as *neighborhood* for i^{th} and j^{th} vehicles and leads to an interconnected graph as follows:[12, 21]

$$\mathbb{G}(t) = \{\mathbb{V}, \mathbb{E}(t)\} \tag{1}$$

where $\mathbb{V}$ is the set of nodes (vehicles) and $\mathbb{E}(t) \subseteq \mathbb{V} \times \mathbb{V}$ is the set of edges (i,j) at time t, with $i, j \in \mathbb{V}$. This graph topology enables one to represent all configurations of the subgroups.

In this paper, the *communication interactions* among the cooperative vehicles have the main importance and hence the presented interaction graph is used to represent the communication interactions and hence is called "*communication graph topology*". Also, it is usually assumed that the "*communication graph topology*" has a particular structure, is fixed, and is set manually before the mission or automatically during the mission. In this paper, the required condition on the *communication graph topology* for communication failure detection is sought.

A. Undirected Communication Graph Topology

If the interaction graph is *undirected* then $(i, j) \in \mathbb{E}$ implies $(j, i) \in \mathbb{E}$ even though it does not appear in $\mathbb{E}$. In fact, the neighborhood relation is necessarily mutual.

The *undirected* communication graph topology is suitable for modeling a bidirectional communication topology. Also N_n^i denotes the number of neighbors of vehicle $i \in \mathbb{V}$ when an *undirected* communication topology is used.

B. Directed Communication Graph Topology

Still if an edge (i, j) connecting the i^{th} node to the j^{th} node is present, it means that the i^{th} and j^{th} vehicles have an interaction and it is said that:

- i^{th} and j^{th} vehicles are *neighboring* vehicles and
- i^{th} vehicle is the *follower* of the j^{th} vehicle and
- j^{th} vehicle is the *leader* of the i^{th} vehicle.

The main distinction between *undirected* and *directed* graph topology is that with the *directed* interaction graph $(i, j) \in \mathbb{E}$ does not imply necessarily $(j, i) \in \mathbb{E}$. Using this flexible graph topology allows representing all interactions of subsystems. Also, let N_l^i and N_f^i denote the number of the *leaders* and *followers* of vehicle $i \in \mathbb{V}$ respectively when a *directed* communication topology is used. This representation does not conflict with *undirected* graph formulation; it is a general form of *undirected* graph topology. The *directed* graph topology allows modeling the unidirectional communication topologies.

IV. Failure Detection Scheme

To monitor the status of the communication devices a "*Healthy*" signal is introduced which is communicated between each pair of neighboring vehicles frequently. The "*Healthy*" signal does not impose considerable communication load but allows providing coordination among vehicles for communication failure detection. Also, the terms *small communication delays* τ_S and *large communication delays* τ_L are introduced as following: let δ denotes the sampling time then $\tau_S < \delta$ and $\tau_L \geq \delta$. The sampling time is chosen as the threshold between the small and large communication delays because most of the decentralized control schemes for cooperative multiple vehicles require the information from neighboring vehicles before any sampling time. It means if the delay of communicated messages is subject to large delays a communication failure is concluded from the control perspective and a reconfigurable fault tolerant controller which relies on the delayed information should be employed. This is the reason why in this paper the small communication delays are considered as the fault-free cases.

In the fault-free situation, at each sampling time every vehicle in the team receives/sends the "*Healthy*" signal from/to their neighbors with a small delay as less than sampling time. Figure 2 shows the inter-vehicle communication between two neighboring vehicles and the information exchanged for fault-free (delay-free) condition. As seen the exchanged messages are not subject to delay.

Figure 2. *The inter-vehicle communications between two neighbors in fault-free condition*

The proposed communication failure detection scheme is based on the fact that the communication failure results in break/delay in the communicated messages and hence if the communication delay of received messages is larger than sampling time, which is the limit between *small communication delays* and *large communication delays*, the occurrence of communication failure is concluded (See Figure 3 and compare with Figure 2). Both faulty vehicle and its neighbors can use this sign to detect the failure. However, depending on the type of communication topology this idea needs to be more expanded to find which vehicle is faulty in the team.

In this section, it is assumed that the simultaneous failures do not happen; instead in Section V the reliability of the presented algorithms against the simultaneous failures is discussed.

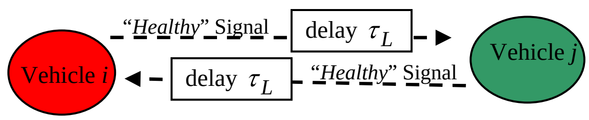

Figure 3. *The inter-vehicle communication between faulty vehicle i and healthy neighbor j*

A. Fault Detection with Undirected (Bidirectional) Communication Graph Topology

When an *undirected* communication topology is used the vehicles are forced to maintain a bidirectional communication structure. Assume at some time the vehicle $i \in \mathbb{V}$ does not hear from its neighbors. The question is that how vehicle i determines whether the break in the messages is due to failure in its own communication device or that of its neighbors. The following theorem is presented to answer this question:

Theorem 1: If an *undirected* communication graph topology is set such that: $\forall i \in \mathbb{V}\,; N_n^i \geq 2$ then the communication fault is detectable and the faulty vehicle in the team can be identified.

Proof: The proof follows from the following statements:

1- *If any vehicle* $i \in \mathbb{V}$ *hears from all neighbors (* $N_n^i \geq 2$ *) without delay then it concludes that neither its communication device nor those of neighbors is faulty.*

2- *If vehicle i does not hear after a certain time (* $\tau_L \geq \delta$ *) from all its neighbors it concludes that its communication device is faulty and does not allow it to communicate with neighbors.*

3- *Accordingly, once the vehicle i hears from at least one neighbor without delay it concludes that its communication device is not faulty, and that the communication device of the neighbor(s) of vehicle i that vehicle i does not hear from it (them) is faulty and does not allow vehicle i to hear from that (them).*

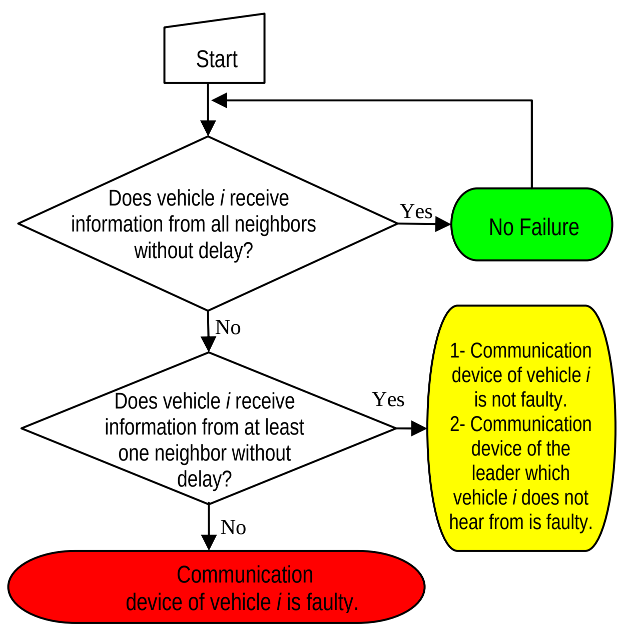

Figure 4. *Fault detection algorithm for vehicle i with undirected communication topology*

In this way the failure is detected and the faulty vehicle is identified by all neighbors. This fault detection algorithm is summarized in Figure 4. This fault detection algorithm monitors the status of the communication device at each sampling time to detect the possible communication faults and the faulty vehicle in the team.

B. Fault Detection with Directed Communication Graph Topology

The main difference between the undirected and directed communication topology is that with the directed graph topology the communication flow may not be necessarily bidirectional between two neighboring vehicles; this implies there may exist some team members which do not receive any information from neighbors as they are only leaders. On the other hand, the proposed fault detection algorithm presented in Section IV.A for *undirected* graph topology requires that each vehicle in the team receives the information from at least two team members. Thus, the algorithm in Section IV.A fails for the case of *directed* communication graph where there may exist some vehicles which are only *leader* and do not receive information from other team members.

In the fault-free condition all the followers receive the "*Healthy*" signal from their leaders with no delay (or a small delay as less than sampling time). If the communication delay of received "*Healthy*" signal is larger than sampling time (execution horizon), which is the limit between *small communication delays* and *large communication delays*, the occurrence of communication failure is concluded (See Figure 5 and compare with Figure 3 and Figure 2). The algorithm for detecting the communication failure in the case of directed communication topology is presented in Figure 6.

Theorem 2: If a *directed* communication graph topology is set such that: $\forall i \in \mathbb{V}\,; N_l^i \geq 2$, then using the detection algorithm presented in Figure 6, the communication failure is detectable and the faulty leader in the team can be identified.

Proof: The proof goes along the statements presented in the proof of *Theorem 1* in Section IV.A for the undirected communication graph.

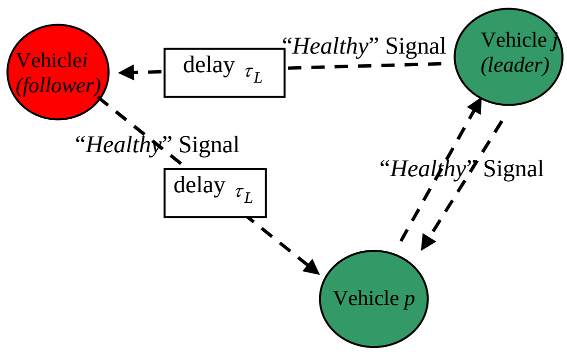

Figure 5. *The inter-vehicle communication between faulty (vehicle i) and healthy (vehicles p and j) vehicles using directed communication topology*

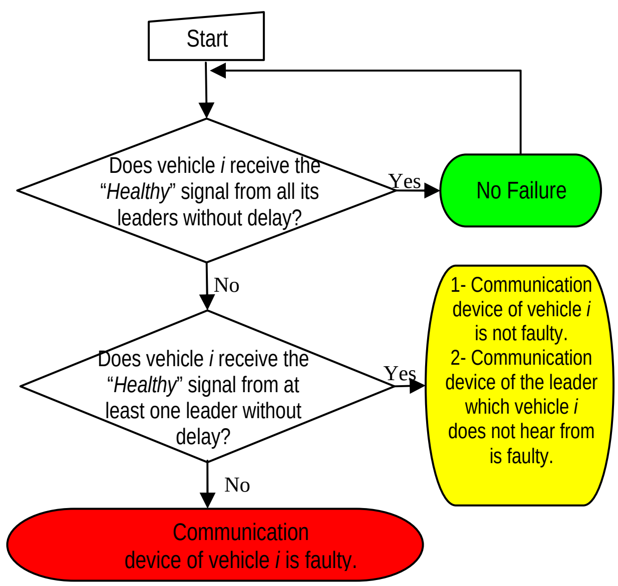

Figure 6. *Fault detection algorithm for vehicle i with directed communication topology*

C. Fault Detection for Communication Devices with Separate Transmitter (TX) and Receiver (RX)

So far, it is assumed that the transmission and receiving the information is performed through the same device for each vehicle and hence the communication device failure implies the failure for both transmitter and receiver; however, if the communication unit have separate receiver (RX) and transmitter (TX) devices[22] (or channels) then the failure may happen to only one of them; this case needs more complex fault detection algorithms.

For instance, if the TX of one vehicle is faulty and the RX is healthy then the faulty vehicle can receive the information from neighbors but it is not able to send the information to neighbors; in this situation if the presented algorithms of previous sections are used the faulty vehicle does not know whether the neighboring vehicles can receive the information or not. Then a suitable algorithm is required to monitor and detect the failure in RX and TX separately.

1. RX Fault Detection

If the RX of one vehicle is faulty then depending on whether the communication topology is directed or undirected any of the algorithms presented in Sections IV.A and IV.B can be used to detect this failure. In this case, the "*Healthy*" signal is used to detect the possible failures in RX.

2. TX Fault Detection

To detect the failure a new "*Acknowledgment*" signal is considered which imposes the neighboring vehicles to acknowledge the receipt of the "*Healthy*" signal, see Figure 7. Any vehicle $j \in \mathbb{V}$ receiving the "*Healthy*" signal from neighbor $i \in \mathbb{V}$ (where $(j,i) \in \mathbb{E}$) will confirm the receipt of the "*Healthy*" signal to the sender $i \in \mathbb{V}$ by sending back the "*Acknowledgment*" signal. If any vehicle $i \in \mathbb{V}$ does not receive the "*Acknowledgement*" signal after two sampling times from at least one of its neighbors then a failure in TX is concluded. The algorithm for detecting the TX failure is presented in Figure 8.

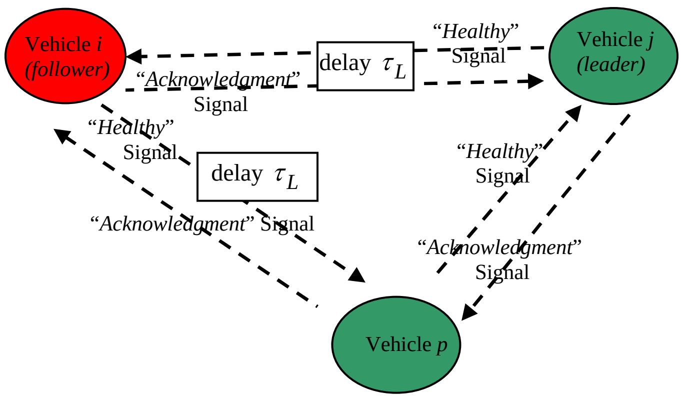

Figure 7. *The inter-vehicle communication between faulty (vehicle i) and healthy (vehicles p and j) vehicles where the transmitter TX and receiver RX communication devices are separate*

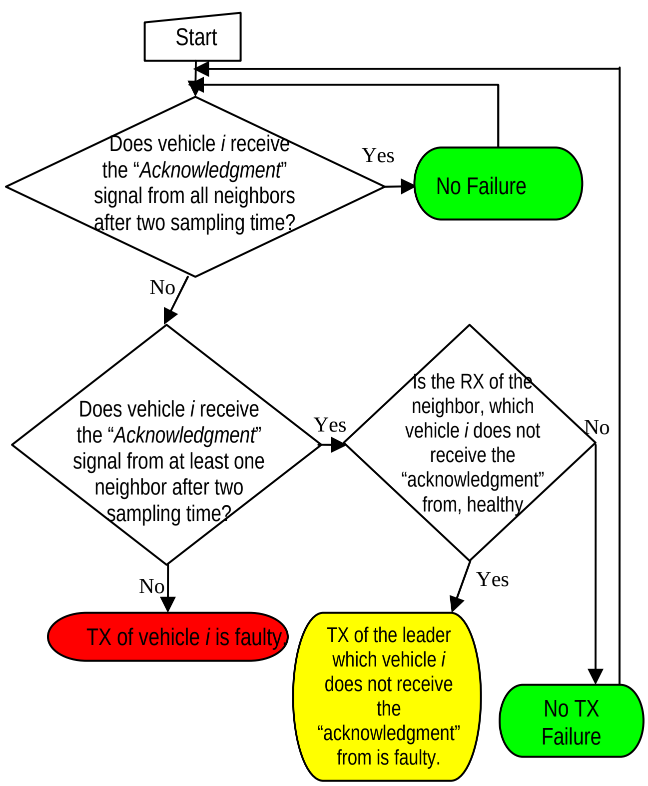

Figure 8. *TX fault detection algorithm for vehicle i with directed or undirected communication topology*

Theorem 3: If a *directed* communication graph topology is set such that: $\forall i \in \mathbb{V}\ ; N_f^i \geq 2$ and the RX is healthy; then using the detection algorithm presented in Figure 8, the TX communication fault is detectable and the faulty leader in the team can be identified.

Proof: The proof goes along the statements presented in the proof of *Theorem 1* in Section IV.A.
The theorem is presented for the *directed* graph since it is a general form of the *undirected* graph.

V. Reliability of Fault Detection Algorithms

The presented fault detection algorithms in this paper for the communication failure may fail if simultaneous failure happens and the communication graph topology is not well-connected. For example, assume vehicle *i* has 2 neighbors; if the communication device of both neighbors become faulty simultaneously, then vehicle *i* does hear from them; therefore, according to algorithm presented in Figure 4, vehicle *i* concludes that its own communication device is faulty; which is a false failure conclusion. In this section, the reliability of the proposed algorithms is investigated through probability analysis.

A. Reliability against Simultaneous Failures

The proposed fault detection algorithms of Sections IV.A and IV.B work properly if the communication device of at least one neighbor is healthy. The first question is: is it possible that the communication devices of all neighbors of vehicle *i* fail simultaneously which leads to false failure conclusion for vehicle *i* about its communication device? The answer is yes; if the probability of failure for one communication device is *p*, the probability for simultaneous failure of communication devices of all neighbors for the case of *undirected* communication graph is calculated as follows:

$$\textit{Probability of Simultaneous Failures} = p^{N_n^i} \tag{2}$$

Then the reliability is:

$$\textit{Reliability against simultaneous failures} = 1 - p^{N_n^i} \tag{3}$$

and likewise for *directed* communication graph:

$$\textit{Reliability against simultaneous failures} = 1 - p^{N_l^i} \tag{4}$$

Eqs. (3), (4) imply that increasing the number of neighboring vehicles enhances the reliability.

B. Reliability against Byzantine Fault

For the case where TX and RX are separate, since the TX fault detection is based on sending the "*Healthy*" signal and receiving the confirmation through the "*Acknowledgment*" signal the Byzantine fault[23,24] may occur. The Byzantine fault is first referenced in the network of computers where the receipt of any message to any destination computer is confirmed through sending back a confirmation signal. However, the confirmation signal needs another confirmation signal from the recipient. This leads to an inconclusive sequence of events and is referred to as the Byzantine fault; for distributed networks a Byzantine fault analysis is required.[24] In the proposed algorithm for TX fault detection presented in Figure 8, false fault detection similar to the Byzantine fault may happen if either TX or RX of the neighbors becomes faulty. Then the reliability of TX fault detection against Byzantine fault is calculated as follows: If $\forall i \in \mathbb{V}$ then let:

$$\textit{Probability of failure of receiver of vehicle i} = P(R^i) \tag{5}$$

$$\textit{Probability of failure of transmitter of vehicle i} = P(T^i) \tag{6}$$

Then for $\forall j \in \mathbb{V}$ and $(i, j) \in \mathbb{E}$, the probability of failure of receiver of *i* or receiver of *j* or transmitter of *j* is:

$$P(R^i) \bigcup P(R^j) \bigcup P(T^j) = P(R^i) + P(R^j) + P(T^j) - P(R^i)P(R^j) - P(R^i)P(T^j) - P(R^j)P(T^j) \tag{7}$$

Hence, for all the followers:

$$\prod_{j|(i,j)\in\mathbb{E}} \left[P(R^i) \bigcup P(R^j) \bigcup P(T^j) \right] = \prod_{j|(i,j)\in\mathbb{E}} \left[P(R^i) + P(R^j) + P(T^j) - P(R^i)P(R^j) - P(R^i)P(T^j) - P(R^j)P(T^j) \right] \tag{8}$$

Therefore:

$$\textit{Reliability} = 1 - \prod_{j|(i,j)\in\mathbb{E}} \left[P(R^i) + P(R^j) + P(T^j) - P(R^i)P(R^j) - P(R^i)P(T^j) - P(R^j)P(T^j) \right] \tag{9}$$

If $P(R^i) = P(T^j) = P(R^j) = p$ then

$$Reliability\ against\ Byzantine\ Fault = 1-\left[3p-3p^2\right]^{N_f^i} \quad (10)$$

C. Reliability against Packet Loss Events

The presented detection algorithms work based on measuring the delay. Also, they assume the communication delay can happen if and only if a communication device failure happens. However, this is not always true as the communication delay can happen due to packet losses which can lead to false detection alarm.

Fortunately, the natures of delay due to packet loss and communication device failure are different; the occurrence of packet loss results in random and discontinuous small communication delays lasting a couple of steps[25]; while a communication device failure leads to large communication delays. Therefore, to make the presented algorithms reliable against such cases, the presented algorithms can be robustified using extra intelligent decision makers which do not conclude the failure detection by observing the first delay event, rather they keep tracking the subsequent packets, if after a couple of time steps no signal is received then a communication failure can be concluded. Meanwhile, from a control perspective if after a while no signal is received due to any reason (communication device failure or packet loss) it is concluded as a faulty situation.

Assume the complementary intelligent algorithm makes the decision based on measurements of a sequence of *m* steps; also, the probability for packet loss at each step is *p*, then for an undirected graph we have:

$$Probability\ of\ Packet\ Loss\ in\ m\ Subsequent\ steps = p^m \quad (11)$$

Hence, $\forall i \in \mathbb{V}$:

$$Probability\ of\ packet\ loss\ in\ m\ subsequent\ steps\ for\ all\ neighbors\ simultaneously = p^{mN_n^i} \quad (12)$$

Therefore:

$$Probability\ against\ packet\ loss\ events = 1-p^{mN_n^i} \quad (13)$$

And likewise for the directed graph topology:

$$Probability\ against\ packet\ loss\ events = 1-p^{mN_l^i} \quad (14)$$

D. Discussions

The eqs. (3), (4), (10), (13) and (14) imply that the reliability decreases dramatically as the number of neighbors increases. For example, if the probability of failure of one vehicle's communication device is p=10%=0.1 and vehicle *i* has $N_n^i = 2$ neighbors, then the probability for simultaneous failure of communication devices of both neighbors of vehicle *i* is 0.1^2=0.01 and the reliability of algorithm is: 1.0-0.01 = 0.99 or 99%. This suggests that for a more reliable fault detection algorithm more vehicles should communicate (cooperate) and exchange the "*Healthy*" and "*Acknowledgment*" signals; in fact, more cooperation leads to more reliability and hence the desired reliability of the algorithms can be achieved by appropriate setting the communication graph topology.

VI. Conclusion

In this paper three algorithms are presented for the communication failure detection for the case of *directed* communication graph, *undirected* communication graph and cases where separate receiver and transmitter devices are used. The proposed algorithms require that each vehicle in the team exchange the "*Healthy*" and "*Acknowledgment*" signals with at least two other neighboring vehicles.

The reliability of the proposed algorithms against simultaneous failures, Byzantine failures and packet loss is also discussed. Some formulas are suggested for calculating the reliability. It is concluded that any arbitrary level of reliability is achievable through setting the *communication graph topology* appropriately.

Acknowledgement

This work was supported in part by National Science and Engineering Research Council of Canada (NSERC), through a strategic project and a discovery project.

References

[1]Chen, J., Kher, S. and Somani, A., "Distributed Fault Detection of Wireless Sensor Networks," *International Conference on Mobile Computing and Networking*, Los Angeles, CA, USA, 2006, pp. 65-72.

[2]Franceschelli, M., Giua, A. and Seatzu, C., “Decentralized Fault Diagnosis for Sensor Networks,” *5th IEEE Conference on Automation Science and Engineering*, Bangalore, India, Aug. 2009.
[3]Sheth, A. Hartung, C. and Han, R., “A Decentralized Fault Diagnosis System for Wireless Sensor Networks,” *2nd Mobile Ad Hoc and Sensor Systems Conference*, Nov. 2005, pp. 192-194.
[4]Wang, P., Zheng, J. and Li, C., “Cooperative Fault-Detection Mechanism with High Accuracy and Bounded Delay for Underwater Sensor Networks”, *Wireless Communications & Mobile Computing*, Vol. 9, No. 2, Feb. 2009, pp 143-153.
[5]Heredia, G., Caballero, F., Maza, I., Merino, L., Viguria A. and Ollero, A., “Multi-Unmanned Aerial Vehicle (UAV) Cooperative Fault Detection Employing Differential Global Positioning (DGPS), Inertial and Vision Sensors,” *Sensors*, Vol. 9, No. 9, Sep. 2009, pp. 7566-7579.
[6]Yang, M.and Fei, Z., “A Cooperative Failure Detection Mechanism for Overlay Multicast,” *Journal of Parallel and Distributed Computing*, Vol. 67, No.6, 2007, pp. 635-647.
[7]Badonnel, R. and Burgess, M., “Fault Detection in Autonomic Networks Using the Concept of Promised Cooperation”, *Book Series: Lecture Notes in Computer Science*, 2007, pp. 62-73.
[8]Cao, J., Sun, Z. and Su, J., “CPS: A Cooperative Probe Based Failure Detection Scheme for Application Layer Multicast,” *International Conference on Electronic Computer Technology*, Macao, 20-22 Feb. 2009, pp. 86-91.
[9]Tino, R., Terra, M. H. and Bergerman, M., “A Fault Tolerance Framework for Cooperative Robotic Manipulators,” *Control Engineering Practice,* Vol. 15, No. 5, 2007, pp. 615-625.
[10]Besada, J. A., Garcia, J., Portillo, J., Molina, J. M., Varona, A. and Gonzalez, G., “Airport Surface Surveillance based on Video Images,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 41, No. 3, July 2005, pp. 1075-1082.
[11]Innocenti, M. and Pollini, L., “Management of Communication Failures in Formation Flight,” *Journal of Aerospace Computing, Information, and Communication,* Vol. 1, No. 1, Jan. 2004, pp. 19-35.
[12]Olfati-Saber, R. and Murray, R. M., “Consensus Problems in Networks of Agents with Switching Topology and Time Delays,” *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, 2004, pp. 1520-533.
[13]Mehra, R. K., Boskovic, J. D. and Li, S. M., “Autonomous Formation Flying of Multiple UCAVs under Communication Failure,” In *IEEE Position, Location and Navigation Symposium,* San Diego, USA, March 13-16, 2000, pp. 371-378.
[14]Ousingsawat, J. and Campbell, M. E. “Optimal Cooperative Reconnaissance Using Multiple Vehicles,” *AIAA Journal of Guidance, Control and Dynamics,* Vol. 30, No. 1, 2007, pp. 122-132.
[15]Weitz, L. A. and Hurtado, J. E., “Investigating Time-Delay Effects for Multivehicle Formation Control,” *AIAA Journal of Aerospace Computing, Information, and Communication,* Vol. 5, No. 9, 2008, pp. 321-336.
[16]Izadi, H., Gordon, B. W. and Zhang, Y. M., “Decentralized Receding Horizon Control of Multiple Vehicles subject to Communication Failure,” the *American Control Conference* , St Louis, USA, June 10-12, 2009.
[17]Langson, W., Chryssochoos, I., Rakovic, S. V. and Mayne, D. Q., “Robust Model Predictive Control using Tubes”, *Automatica,* Vol. 40, No. 1, 2004, pp. 125-133.
[18]Zhang, X., Xu, R., Kwan, C., Haynes, L., Yang, Y. and Polycarpou, M., “Fault Tolerant Formation Flight Control of UAVs,” *International Journal of Vehicle Autonomous Systems*, Vol. 2, No. 3/4, 2004, pp. 217-235.
[19]Cheng, L., Wang, Y. and Zhu, Q., “A Communication-Based Multi-Robot Rigid Formation Control System: Design and Analysis,” *Int. J. Modeling, Identification and Control*, Vol. 1, No. 1, 2006.
[20]Izadi, H., Gordon, B. W. and Rabbath, C. A., “Decentralized Receding Horizon Control Using Communication Bandwidth Allocation,” *Proceedings of the 47th IEEE Conf. on Decision and Control*, Cancun, Mexico, Dec. 9-11, 2008.
[21]Keviczky, T., Borrelli F. and Balas, G. J., “Stability Analysis of Decentralized Receding Horizon Control for the Decoupled Systems,” *Proc. of 44th IEEE Conference on Decision and Control, and the European Control Conference,* Seville, Spain, Dec. 12-15, 2005, pp. 1689-1694.
[22]Shannon, C. E., “A Mathematical Theory of Communication,” *Bell System Technical Journal*, Vol. 27, July, 1948, pp. 379-423.
[23]Lamport, L., Shostak, R. and Pease, M., "The Byzantine Generals Problem," *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3, July 1982, pp, 382-401.
[24]Miguel C. and Liskov, B., “Practical Byzantine Fault Tolerance,” the *3rd Symposium on the Operating Systems Design and Implementation*, New Orleans, USA, February 1999.
[25]Huanga, M. and Dey, S., “Stability of Kalman Filtering with Markovian Packet Losses,” *Automatica*, Vol. 43, 2007, pp. 598 – 607