Formal Verification of Hierarchically Distributed Agent Based Protection Scheme in Smart Grid

Shravan Garlapati and Sandeep K. Shukla

Virginia Polytechnic and State University, Blacksburg 24060, USA,
{gshra09,shukla}@vt.edu

Abstract. In recent studies, hierarchically distributed non-intrusive agent aided transmission line distance relaying protection scheme has been proposed. This scheme is meant to provide the distance relays with situational awareness and improve their robustness against hidden failures. Distance relaying protection scheme is a part of safety critical cyber physical system (in particular, power system) and it operates with stringent timing requirements to remove the faulted line out of service. Before putting into practice, it is better to formally verify that the agent based relay supervisory scheme meets the specifications and its usage gives intended results and doesn't carry any negative side effects. Therefore, in this paper agent based relay supervision scheme is formally modelled, validated and its properties are verified using UPPAAL - a timed automata based formal verification tool.

1 Introduction

Power grid transmission lines are usually protected by distance relays which comprises of local primary relays (Zone 1), secondary relays (Zone 2) and remote back up relays (Zone 3) [8]. The main objective of the protection system is to isolate a fault as soon as possible to minimize the negative impact of the fault on the grid and also to minimize the amount of load shed because of the relay induced disconnection of lines. Remote back up relay is preferred to a local backup as a local backup shares the same electrical and communication infrastructure with primary relay; hence vulnerable to "Common Mode Failure" [4][6]. Remote back up relay and primary relay are usually located in different substations and thus are less vulnerable to "Common Mode Failures". Compared with primary relays, remote backup relays operate with longer fault clearing times and also its operation to remove a fault may lead to larger area of load shedding. Therefore, transmission line distance relaying protection system is designed in such a way that the remote back up relay doesn't trip unless it is absolutely necessary i.e. when both zone 1 and zone 2 relays fail or their associated sensors or breakers simultaneously fail to clear the fault. After a thorough analysis of historical blackouts such as 1965 Great North-east blackout, 1977 New York blackout and the 1996 western blackout, North American Electric Reliability Council (NERC) concluded that a Zone 3 relay mis-operation is one of the major causes of cascading outages leading to blackout events [10]. Horowitz et. al. reanalyzed

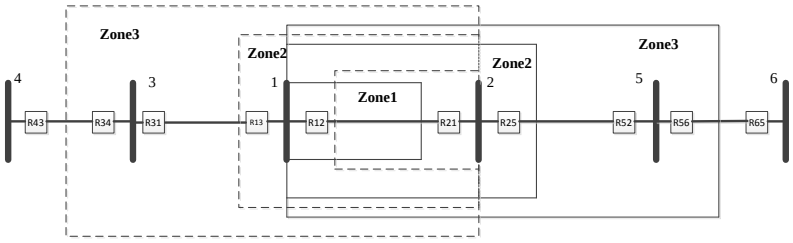


Fig. 1. Zones of Protection

the distance relaying protection scheme and concluded that Zone 3 relay cannot be abandoned as its absence will put a power system at risk [4].

Zone 3 relays can incorrectly trip a line due to hidden failures [11][7]. A hidden failure is a defect (incorrect relay setting or software or hardware error) in a relay which may go unidentified for a long time and gets excited by another event leading to erroneous removal of circuit elements [9]. Because of hidden failures, Zone 3 relays may be extra sensitive to temporary line overloading due to transients, mistake it as a fault in a line and mis-trip even though it is not recognized as a faulty condition by Zone 1 or Zone 2 relay. At this instance, if power system is operating under stressed conditions, the hidden failure induced Zone 3 relay mis-trip may initiate other line trips leading to catastrophic failures like blackouts. One of the main objectives of the smart grid is that the power system will be enabled with communication and networking infrastructure to an unprecedented level, and wide area measurements and controls will provide the power system (transmission and distribution) with unprecedented robustness and prevent untoward incidents such as blackouts. In [1] an agent based relay supervision scheme is proposed to reduce the probability of hidden failure induced trips. Agents are hierarchically distinguished as master and slave agents. The real time communication between the master and slave agents aid Zone 3 relays to classify a fault as a true fault or a hidden failure induced fault and respectively to trip or not to trip.

In this paper we use UPPAAL - a formal verification tool to formally verify and validate the agent based distance relaying protection scheme. Because of strict timing requirements (section II) for the proper functioning of distance relays, time based formal models are needed, and UPPAAL allows us to model these in the form of timed automata, and allows model checking of timed properties on the models. To the best of our knowledge this is the first instance of applying formal verification to the protection scheme in a power system. Remainder of the paper is organized as follows. Section 2 explains distance relaying protection scheme. Summary of related research work is provided in section 3. Section 4 explains modelling of agent based distance relay protection scheme in UPPAAL. Section 5 discusses verification results and some observations are discussed in section 6. Section 7 concludes the paper.

2 DISTANCE RELAYING PROTECTION SCHEME

Distance relays operate based on the principle of impedance ratio, which is the ratio of the magnitude of voltage to that of the magnitude of current. The current and the voltage values measured respectively by current transformer (CT) and voltage transformer (VT) are communicated to the relay. With the current and voltage values as input, relay executes the relaying algorithm and concludes about the presence of a fault. If there is a fault then relay communicates with the breaker to trip the line out of service. If there is no fault, relay repeats the above procedure with the next set of current and voltage values. To account for the inaccuracies in sensing equipment (CT and VT), uncertainty in distance setting of relays and to make sure that there are no blind spots, multiple zones of protection (Zone 1, Zone 2) are employed for each transmission line. In the presence of a fault if the breaker associated with the Zone 1 or Zone 2 relay doesn't trip (due to a failure in CT, VT, relay or breaker), faulted line cannot be isolated from the system. Therefore a backup relay or Zone 3 relay is placed in the remote substation (bus). Thus there exists three different zones of protection i.e. Zone 1, Zone 2 and Zone 3 relays protecting a transmission line. It is already explained above that the remote back up relay is preferred to a local backup relay as the latter can be a victim of "common mode failure" along with the primary relay (Zone 1). Please refer to fig 1 to see a pictorial representation of zones of protection.

Each transmission line is protected by relays at both ends. This is as shown in 1. In order to remove a faulted line out of service relays at both ends should trip. Zone 1 relay operates instantaneously i.e. within 2 cycles (32 msec). A coordination delay of 20 cycles (300 msec) is allowed before Zone 2 relay operates. Zone 3 relay or remote back up relay is allowed to operate with a coordination delay of 1 sec (1000 msec). Coordination delays not only provide selectivity in isolating a faulted section but also ensure reliability of operation of the distance protection scheme [5]. Detailed explanation of zones of protection is out of the scope of this paper. Interested readers are referred to [8].

3 PREVIOUS WORK

This section provides a brief summary of related previous research work. In [1] an agent based Zone 3 relay supervision scheme is proposed to reduce the probability of hidden failure induced trips. As explained in section II, a fault in a single transmission line can be sensed by multiple relays under different zones of protection. In the proposed scheme each relay is associated with an agent(slave) which has the ability to sense and communicate fault status information to other agents. Fault status indicates if there is a fault in the transmission line protected by the relay or not. Based on the responsibilities assigned to them agents are hierarchically distinguished in a master/slave relationship. At any given instance the master agent has the complete information about the fault status sensed by all the relays (communicated by agents) protecting a transmission line. Whenever

a relay senses a fault, its associated slave agent records it and queries the master agent to find out if the sensed fault is a true fault or a falsely perceived fault. Master agent compares the queried slave agent relay's fault status with the other slave agent relays protecting the same transmission line. In order to perform this comparison, master agent must know ahead of time which set of relays are protecting the transmission line. [3] provides an algorithm for the master agent to find out the set of relays protecting a transmission line. If majority of the other relays also sense fault, master agent classifies the fault as a true fault and acknowledges the queried slave agent relay to trip. On the other hand if majority of the other relays protecting the transmission line doesn't sense a fault, master agent categorizes the condition as non-faulty and sends a message to the Zone 3 slave agent relay not to trip. Thus, with the help of agent communication a relay can distinguish a true fault from a hidden failure induced fault. The entire process of sensors sensing the current and voltage values, relay algorithm execution to find the existence of fault, slave agent recording a fault and querying the master agent, master agent comparing fault statuses of different relays protecting a transmission line and acknowledging the queried slave agent relay has to be finished within the relay fault clearing time i.e. 1 sec, 300 msec and 32 msec respectively for Zone 3, Zone 2 and Zone 1 relay. With the current state of the art communication and networking technologies it may be difficult to meet the timing requirements of Zone 1 and Zone 2 relays but Zone 3 relay time constraint may be met. Therefore we restrict our analysis to Zone 3 relay supervision i.e. only Zone 3 slave agent relay queries are answered by the master agent.

It is possible that a larger bus system is geographically wide spread around 100's to 1000's of miles. If a single master agent is employed to serve queries from all the Zone 3 slave agents in such a large power grid, the round trip communication delay over large distances can exceed the Zone 3 fault clearing time. This deceives the purpose of the agent based distance relaying protection scheme. To overcome this issue, in [2] a methodology is provided to divided a power system network into sub-networks with the objective of minimizing the number of master agents required to serve queries from all the Zone 3 slave agents such that the round trip communication timing requirements are met.

4 MODELLING BEHAVIOUR

In UPPAAL the model of system's behaviour is expressed as the composition of the behaviour models of its individual components. The main components of the agent aided distance relaying protection scheme are sensors (CT and VT), Zone 1 relay, Zone 2 relay, Zone 3 relay, breakers, slave and master agents. There exist two different models for both the sensor and the breaker. The reason for this is as follows: Zone 3 relay operates when both Zone 1 and Zone 2 relay fail and/or their associated both breakers or sensors fail simultaneously. Practically it is possible that the Zone 3 relay and its sensor and breaker equipment can fail but we didn't consider this scenario in our model. The main reason being the probability of Zone 1, Zone 2 and Zone 3 relays failing simultaneously is very low.

Moreover Zone 3 is the only backup available. If we consider the case where Zone 3 relay also fails along with Zone 1 and Zone 2 relay we cannot successfully verify the distance relaying scheme. Hence the case of either Zone 3 relay or its sensor or breaker failure is not considered. Therefore the Sensor 1 and the Breaker 1 models have *failed* state whereas the Sensor 2 and the Breaker 2 models do not have *failed* state. Sensor 1 and Breaker 1 model the behaviour of the sensor and the breaker associated with Zone 1 and Zone 2 relays. Whereas the behaviour of the sensor and the breaker of the Zone 3 relay are respectively presented in the Sensor 2 and the Breaker 2 models.

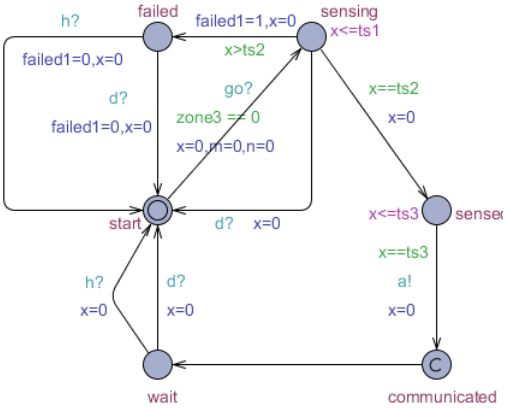


Fig. 2. Sensor1 automaton

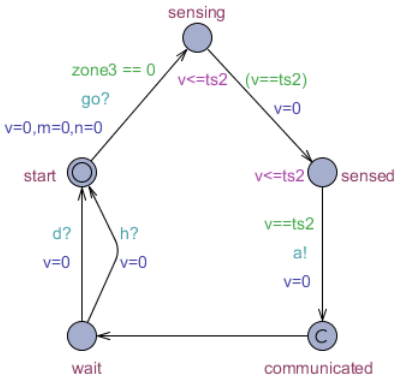


Fig. 3. Sensor2 automaton

A reset transition moves an automaton from any state to the *start* state. The following two reset transitions are used by all the automatons described in this section. These two reset transitions are used multiple times to explain the behaviour of all the automata. Instead of rewriting these transitions many times they are just explained once here. At this point they may or may not be clearly understood but by the end of this section their relevance should become apparent.

1. *Reset I*: In our system model, when breakers at both ends of the line trip, a reset signal is sent via the broadcast channel d to all automatons to move to the *start* state. As shown in Fig 1 a transmission line is protected by Zone 1, Zone 2 and Zone 3 relays in both the directions. At least two relays i.e. one relay per direction have to trip in order to remove a faulted line out of service. In total there are at least six relays protecting a transmission line. Depending on which relay out of these six relays trip's last, any of the six breakers can send a reset signal via the urgent broadcast channel d to reset the whole system.
2. *Reset II*: If any of the Zone 3 relay senses no fault, then it moves from the location *calculate* to *nofault* transmitting a *nofault* signal via the broadcast

channel *nf*. The remaining Zone 1, Zone 2 and Zone 3 relays receive the signal via the broadcast channel *nf* and move to *nofault* state. Zone 3 relay moves from *nofault* to *start* state by transmitting a reset signal on the broadcast channel *h*. All the automata move from their current location to *start* state after receiving a reset signal on the broadcast channel *h*.

In the following description of timed automata words "state" and "location" are used interchangeably and they both mean the state of an automaton.

a) Sensor 1: Both the current and the voltage transformer are modelled as a single sensor. Timed automaton model of Sensor 1 is as shown in figure 2. It uses the clock x to measure time. Sensor 1 moves from initial state *start* to the *sensing* state via the urgent channel *go* if the boolean variable *zone3==0*. If *zone3==0* it is an indication that all automatons are in start state. Also during this transition integer variables m and n are set to zero. These two variables are used by the master agent. If Sensor 1 is functioning correctly, it senses the current and the voltage values within *ts2* msec and moves from the state *sensing* to *sensed*. On the other hand if Sensor 1 is malfunctioned it will make a transition from the *sensing* state to the *failed* state in the time interval *(ts2,ts1)*. Automaton can move from *failed* to *start* state via *Reset I* or *Reset II*. If Sensor 1 makes transition from the location *sensing* to *sensed*, within *ts2* msec it sends the voltage and the current values to the respective relay via the synchronization channel a and moves to the committed location *communicated*. In networks of timed automata describing a system if any automaton is in a committed location next transition is from that location. Committed location is used in the execution of atomic sequence. From the committed location *communicated*, Sensor1 makes a transition to the *wait* state via the urgent channel *go*. Automaton can move from the *wait* to *start* state via *Reset I* or *Reset II*.

b) Sensor 2: The behaviour of the Sensor 2 is similar to that of the Sensor 1 except that the former doesn't have the *failed* state. Timed automaton of the Sensor 2 is as shown in figure 3.

c) Zone 1 Relay: Timed automaton of Zone 1 Relay is as shown in figure 4. It uses the clock z to measure time. It receives the current and the voltage values from the Sensor 1 via the synchronization channel a and moves to the *calculate* state from the *start* state. Relay consumes *tz2* to *tz3* msec of processing time to find out if the transmission line protected by it is faulty or not. If there is a fault in the transmission line, Zone 1 relay moves from the state *calculate* to *faulty* within the time interval *(tz2,tz3)*. During this transition boolean variables *fault1, fault2* are set and a fault signal is sent to the slave agent associated with the relay via the communication channel e. It is aforementioned that in order to remove a faulted line out of service relays at both ends of the line have to sense and respective breakers have to trip. The Boolean variables *fault1* and *fault2* provide the fault status of the relays at both ends of the line. Initially both *fault1* and *fault2* are set to zero indicating a no fault condition. Whenever system (relay) senses a fault both *fault1* and *fault2* are set to one. The breaker trip at one end of the line resets *fault1* while the breaker trip at the other end of the line sets *fault2* to zero which removes the fault from the system.

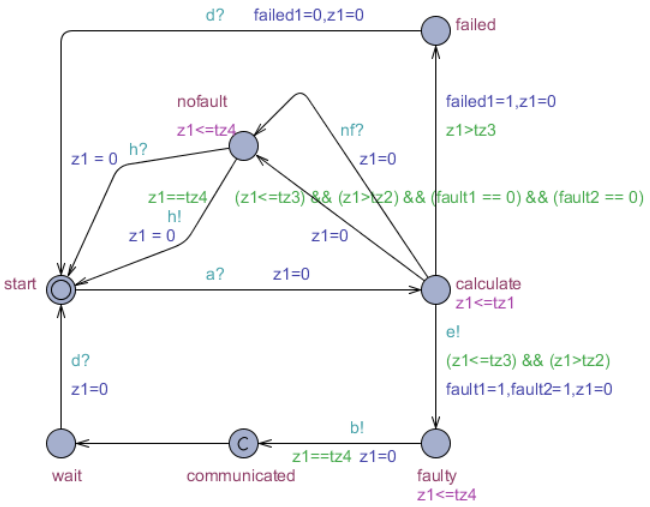


Fig. 4. Zone 1 Relay automaton

On the flip side if there is no fault then relay makes a transition from the state *calculate* to *nofault* via *Reset II* if the Boolean variable *fault1* is not set. If *fault1* is already set, it is an indication that the system has already detected the fault via Zone 2 or Zone 3 relay and Zone 1 relay cannot move to *nofault* state. Zone 1, Zone 2 and Zone 3 relays all operate simultaneously until fault detection stage and detect the fault within the time interval *(tz2,tz3)*. The main difference in Zone 1, Zone 2 and Zone 3 relay models is the time instance at which they send tripping signal to their respective breakers. So, in the presence of a transmission line fault it is hard to predict which relay can first detect that fault and set the *fault1* variable. Thus it is necessary to check if *fault1* is set before moving from the state *calculate* to the state *nofault*. If Zone 1 Relay is malfunctioned, it does not respond in the time interval (tz2,tz3) and makes a transition to the *failed* state in time interval (tz3,tz1) and sets the Boolean variable *failed1* giving an indication that the relay has failed. Relay makes transition from the *failed* state to the *start* state when it receives a reset signal on the broadcast channel *d* via *Reset I*. During this transition it resets the Boolean variable *failed1*. A communication delay of tz4 msec is involved in sending a trip signal from a relay to the breaker. So, with a delay of tz4 msec Zone 1 Relay automaton moves from the state *faulty* to *communicated* sending a trip signal to its breaker via the synchronization channel *b*, then moves to the *wait* state via the urgent channel *go*. Automaton then makes a transition from the *wait* to the *start* state via *Reset I*.

d) Zone 2 Relay: Timed automaton of Zone 2 Relay is as shown in figure 5. It uses the clock z_2 to measure time. Transitions from the state *start* to *faulty* are similar to that of Zone 1 Relay with few changes. First, the Boolean variable

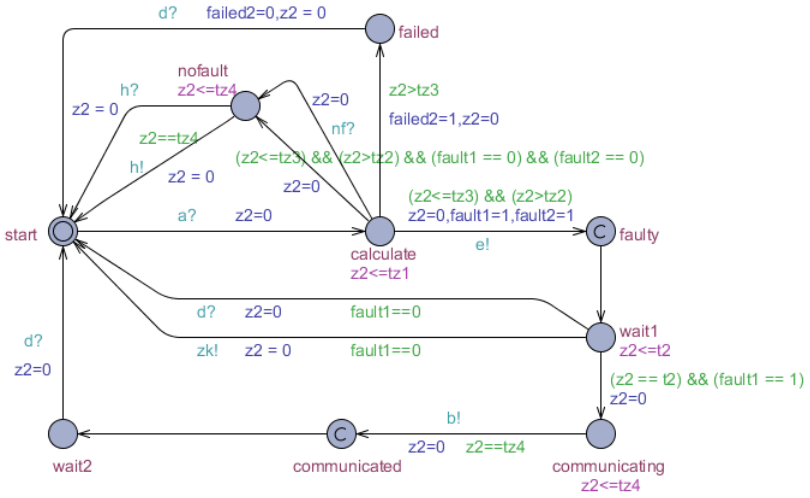


Fig. 5. Zone 2 Relay automaton

failed1 is replaced with the Boolean variable *failed2*. Second, the *faulty* state is a committed location. From the *faulty* state automaton makes a transition to the *wait1* state. If the breaker of Zone 1 Relay trips and resets the *fault1* variable, it is an indication that the system is fault free and Zone 2 relay moves to *start* state in time interval $[0, t2)$ msec by transmitting a signal via the urgent channel *zk* or by receiving a *Reset I*. If the breaker associated with Zone 1 relay doesn't trip within a communication delay of t_2 Zone 2 Relay moves to the *communicating* state from the *wait1* state to send a trip signal to its breaker. Zone 2 Relay automaton makes a transition from the *communicating* to the *communicated* state with a delay of tz4 msec. During this transition Zone 2 Relay sends a trip signal to its breaker via the synchronization channel *b*, and then moves to the *wait2* state via the urgent channel *go*. Automaton then makes a transition from *wait* to the *start* state via *Reset I*.

e) Zone 3 Relay: Timed automaton of Zone 3 Relay is as shown in figure 6. It uses the clock z_3 to measure time. Zone 3 relay automaton behaviour is almost similar to that of Zone 2 Relay, except that it doesn't have *failed* state. The other change is that the variable $t2$ is replaced with the variable $t3$. When Zone 3 relay is in *wait1* state, within the coordination delay $[0, t3)$ if the breaker associated with either the Zone 1 or Zone 2 relay doesn't trip, Zone 3 relay moves to the *communicating* state and sends a signal to its breaker to trip the line out of service.

f) Breaker 1: Timed automaton of Breaker 1 is as shown in figure 7. Automaton uses the clock y to measure time. Breaker 1 is initially in the *start* location. Automaton receives a trip signal from its associated relay via the channel *b* and makes a transition from the location *start* to *received*. After receiving the trip

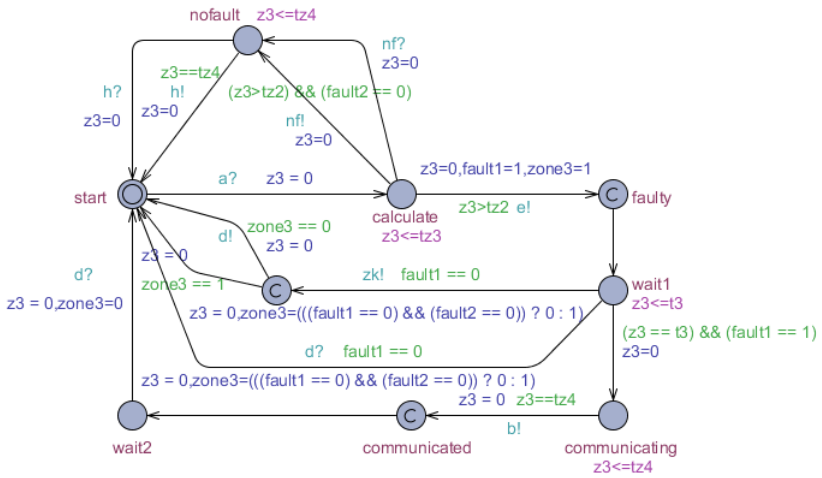


Fig. 6. Zone 3 Relay automaton

signal from the relay, breaker and its associated electromechanical machinery trips a line out of service with a delay of tb1 msec. So, assuming that the breaker is functioning correctly automaton moves from the state *received* to the committed location *intermediate* in tb1 msec and resets the *fault1* variable, indicating that the line is tripped. If a transmission line is faulty then breakers at both ends of the line have to trip to remove the line out of service. So, to make a transition from the state *intermediate* to *tripped*, automaton performs a check to find out if the breaker on the other end of the line has tripped or not. If it is tripped *fault2* is reset otherwise *fault2* is set. Irrespective of whether *fault2* is set or reset automaton makes a transition from location *intermediate* to *tripped*. But if *fault2* is reset, automaton while making a move from the state *intermediate* to *tripped* transmits a faultfree signal via the channel c to the observer automata, giving an indication that the system is free of fault. When automaton is in *received* state, if the breaker doesn't respond for more than tb2 msec then it moves to *failed* state in time interval $[tb2, tb1)$. From the *failed* state breaker can make transition to the *start* state via *Reset I*. Transition from the location *tripped* to *start* occurs when both the Boolean variables *fault1* and *fault2* are reset. Also during this transition automaton sends a reset signal via *Reset I*.

g) Breaker 2: Timed automaton of Breaker 2 is approximately similar to that of Breaker 1 with the only change being Breaker 2 doesn't have a *failed* state. It is as shown in figure 8.

h) Observer automaton (OA): Observer automata captures the high level behaviour of the distance relaying protection scheme i.e. whether the system is *faultfree* or *faulty*. As shown in figure 9(a) observer automata has only 2 states i.e. *faultfree* and *faulty*. Automaton is initially in *faultfree* location. When Zone 1 or Zone 2 or Zone 3 Relay senses fault, they transmit fault signal n channel *e*.

OA listen's it and moves to the *faulty* location. Transition from the state *faulty* to *faultfree* occurs when the automaton receives a reset signal via *Reset I*.

i) Helper automaton: As shown in figure 9(b) helper automaton has two transitions and one state. Whenever any automata has to make an urgent transition, helper automata sends a signal via the urgent channel *go* and other automata listens and makes a transition. Similarly Zone 2 and Zone 3 Relay make an urgent transition from the state *wait1* to the *start* state via the urgent channel *zk*.

j) Slave agent: Similar to the sensor and the breaker models there exist two different models for the slave agent. Slave agent 1 is used to model the behaviour of the agent located at Zone 1 and Zone 2 relay. Timed automaton of the slave agent 1 is as shown in figure 10(a). Slave agent 1 records the outcome of the relay execution algorithm, records it and reports it to the master agent so that the latter's database is up to date. The current state of the art relays can communicate at 30 times/sec i.e. they can transmit new fault status every 33 msec. Therefore the master agent receives a new fault status from a slave agent 1 every $delay1 = 33$ msec. In our model we declared a global variable *afault* for each slave agent and it is updated with a delay of 33 msec. As slave agent's *afault* variable is declared as global, master agent also has access to it. By declaring *afault* variable as global, model is simplified as fault status value passing is avoided between the master agent and the slave agent.

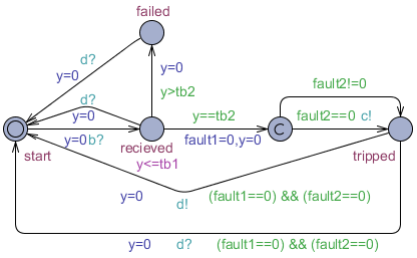


Fig. 7. Breaker1 automaton

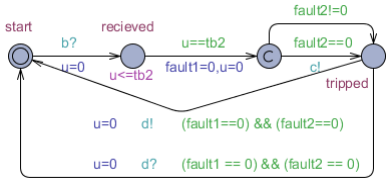


Fig. 8. Breaker2 automaton

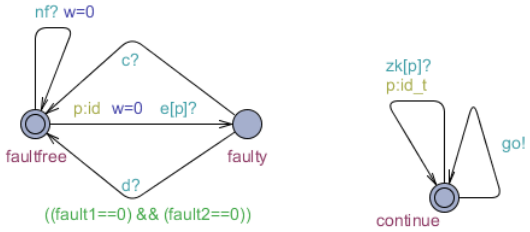
Timed automaton of Slave agent 2 is as shown in figure 11. The behaviour of the agent associated with Zone 3 relay is modelled by the Slave agent 2. Slave agent 2 makes a transition from the state *start* to *received1* after receiving a fault signal from the Zone 3 relay. Similar to the Slave agent 1 fault variable *afault* of the Slave agent 2 is updated with a delay of $delay1 = 33$ msec. Automaton moves from the location *received1* to the committed location *sent* with a delay of 33 msec and sends a fault status update signal via the synchronization channel f to master agent. Then Slave agent 2 makes a transition from the committed location *sent* to the normal location *wait*. In the *wait* state automaton waits for the reply from master agent to confirm if the fault sensed by the Zone 3 relay associated with the Slave agent 2 is a true fault or a hidden failure induced fault.

When automaton is in the *wait1* state, there is a possibility of three different transitions.

1. If either the Zone 1 or Zone 2 relay clears fault, Slave agent 2 makes a transition to the *start* state. When the master agent sends a signal to the Slave agent 2 via the synchronization channel g about whether the fault is a true fault or a hidden failure based fault, Slave agent 2 listens and moves from the location *wait* to the committed location *received 2*. Slave agent 2 then moves from the committed location *received2* to *start*. During the transition from the state *wait* to *received2* boolean variable *fault1* is updated by *function1()*. If the fault is a hidden failure induced fault, *function1()* resets *fault1* variable and if the fault is a true fault, *fault1* variable is set. It is aforementioned that a fault in a transmission line can be sensed by atleast six different relays. As each relay has a slave agent associated with it, atleast six slave agents report to the master agent about the fault status in a line. In *function1()* boolean variable *afault* of Slave agent 2 is compared with *afault* variables of five other slave agents. If at least half (3 out of 6) *afault* variables are set to 1, it is an indication that the transmission line is faulty and the boolean variable *fault1* is set to 1 and the breaker associated with the Zone 3 Relay can trip if both the Zone 1 and the Zone 2 relay breakers fail to trip. On the other hand if more than half (>3 out of 6) of the *afault* variables are set to zero, it is an indication that there is no fault in the line then the Boolean variable *fault1* is set to zero and it is not required for the Zone 3 relay's breaker to trip. If the sensor or the relay fails the respective slave agent's *afault* variable is not taken into consideration in the above decision making which is implemented by *function1()*.
2. If Slave agent 2 is waiting for an acknowledgement from the master agent, it is possible that a breaker associated with the Zone 1 or Zone 2 relay to trip. Therefore it is not required by the slave agent 2 to wait for the fault classification signal from master agent. In this case the transition from *wait* to *start* can occur in two different ways. If Zone 3 relay interprets that either the Zone 1 or Zone 2 relay has tripped and *fault1==0*, slave agent 2 receives a reset signal on the urgent broadcast channel zk and it moves from *wait* to *start* state. This is known as *Zone 3 Reset*. item The transition from *wait* to *start* state can occur via *Reset I*.

k) Master Agent: The behaviour of the master agent is modelled using two timed automatons. The master agent stores the requests in a queue as they are received and processes them based on the first in first out (FIFO) order. As shown in figure 10(b) the master agent receives requests from Slave agent 2 via the synchronization channel f and appends it to queue using the *enqueue()* function. The received request is processed by the master agent task execution timed automaton shown in figure 12. Whenever a request is received, length of the queue *len* is greater than zero and automaton moves from the initial location start to evaluate. There are three possible transitions from *evaluate* state:

1. While Slave agent 2 of Zone 3 relay is waiting to receive a trip/no trip signal from the master agent, breakers associated with either the Zone 1 relay or



(a) Observer Automaton (b) Helper automaton

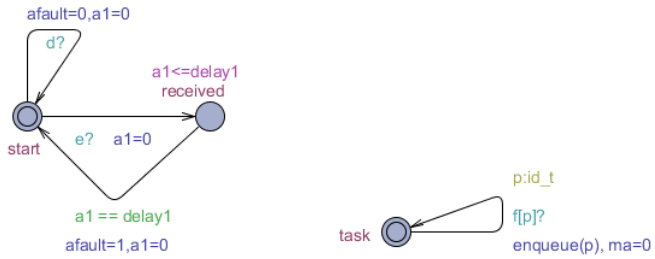
Fig. 9. Different Automatons 1

Zone 2 relay at both ends of the line may trip and reset *fault1* and *fault2* variables. In this case the master agent deletes from its queue the Zone 3 relay Slave agent 2 queries at both ends of the line and moves from the *send* to *start* state.

2. A slave agent is capable of transmitting new fault variable every 33 msec. Therefore a maximum delay of 33 msec is allowed for master agent to process a request. Also, a database query time of 100 msec is assumed in OPNET simulations. A detailed justification is provided in [2] for the selection of database query and master agent service time. Hence the total master agent delay in processing a single query is 133 msec. Therefore the master agent automaton moves from *evaluate* to *send* state approximately in 133 msec.
3. The third possible transition is from *evaluate* to *start* state via the *Zone 3 Reset*.

Automaton can move from the *send* state to *start* state via four different transitions.

1. Within *delay1* msec, master agent processes the next query in queue and sends a reset signal on the channel *g*.
2. The second possible transitions is via *Reset I*.
3. The other two possible transitions are due to *Zone 3 Reset*.



(a) Observer Automaton (b) Helper automaton

Fig. 10. Different Automatons 2

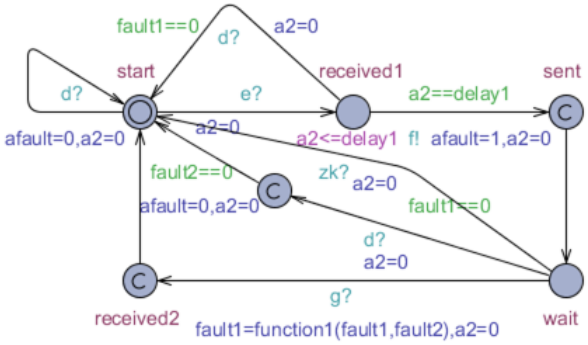


Fig. 11. Slave agent 2 automaton

Table 1. Timing values

Parameter	Before Scaling	After Scaling
ts1	2	-
ts2	1	-
tb1	21	-
tb2	20	-
tz1	8	-
tz2	2	-
tz3	6	-
tz4	4	-
t2	268	33
t3	968	121
delay	133	17
delay1	75	10

5 VERIFICATION

The description of the complete model can be downloaded from www.filebox.vt.edu /users/gshra09/agents.zip. This section explains the properties of the agent based Zone 3 relay supervision scheme that are verified. In UPPAAL Timed Computation Tree Logic (TCTL) is used to specify system properties. Sensor1t(2t,3t) and Breaker1t(2t,3t) are the sensor and the breaker associated with the Zone1t (Zone2t,Zone3t) relay protecting the line at one end whereas Sensor1f(2f,3f) and Breaker1f (2f,3f) are the sensor and the breaker associated with the Zone1f(Zone 2f,Zone3f) relay protecting the line at the other end. afault[0],afault[1],afault[2],afault[3],afault[4] and afault[5] are fault status recorded by slave agents of Zone1t, Zone1f, Zone2t, Zone2f, Zone3t and Zone3f respectively. The following properties are verified.

Safety Property:
a) *A [] no deadlock* i.e. system is deadlock free.
Bounded Liveness Property:
b) *System.faulty* $\longrightarrow$ *((System.faultfree) and (System.w* $\leq$ *153))* i.e. System is fault free within 153 msec. If there is a large range in timing, UPPAAL leads to state space explosion. Therefore timing values in slave agent to master agent communication, Zone 2 and Zone 3 waiting times are scaled by a factor of 8. Actually the system should be fault free within the Zone 3 fault clearing time of 1 sec. A Zone 3 slave agent should receive a response from master agent within 968 msec, scaling this by 8 results in 121 msec. Remaining time of around 32 msec is lost in communication delays between sensor and relay, relay and breaker. These values are not scaled as they are low. Not scaling these values doesn't have any effect on the scaled agent communication delays. Hence the total time available for the system to be fault free is $(121 + 32 = 153msec)$. The timing values before and after scaling are as shown in Table 1.
Model Correctness Properties:
c) *((Sensor1t.failed or Z1t.failed or Breaker1t.failed) and (Sensor2t.failed or Z2t.failed or Breaker2t.failed) and (afault[4] == 1) and (n < m))* $\longrightarrow$ (not (Breaker3f.tripped)). Here 'n' is the number of slave agents with *afault = 1* and 'm' is the number of slave agents with *afault = 0*. If Sensor1t or Zone1t relay or Breaker1t failed and Sensor2t or Zone2t or Breaker2t failed and Zone3t relay slave agent's Boolean variable *afault[4]* is set to 1 then Zone 3 breaker cannot trip if $n < m$.
d) *((Sensor1f.failed or Z1f.failed or Breaker1f.failed) and (Sensor2f.failed or Z2f.failed or Breaker2t.failed) and (n < m) and (afault[5] == 1))* $\longrightarrow$ *(not (Breaker3t.tripped))*.
This property is similar to property c) but this is verified at the other end of the line.
e) *((Sensor1t.failed or Z1t.failed or Breaker1t.failed) and (Sensor2t.failed or Z2t.failed or Breaker2t.failed) and (afault[4] == 1) and (n* $\geq$ *m))* $\longrightarrow$ *Breaker3t.tripped*.
If Sensor1t or Zone1t relay or Breaker t1 failed and Sensor2t or Zone2t or Breaker2t failed and Zone3t relay slave agent's Boolean variable afault[4] is set to 1 then Zone 3 breaker can trip if $n \geq m$ and both n is greater than one. $(n > 1)$ indicates that atleast one relay (Zone1 or Zone2 or Zone3) from both ends of line sense that there is a fault.
f) *((Sensor1f.failed or Z1f.failed or Breaker1f.failed) and (Sensor2f.failed or Z2f.failed or Breaker2f.failed) and (afault[5] == 1) and (n* $\geq$ *m) and (n* $>$ *1))* $\longrightarrow$ *Breaker3f.tripped*. This property is similar to e) but this is verified at the other end of the line.

The main aim of the agent based distance relaying scheme is to aid Zone 3 relays to prevent hidden failure induced trips. Properties *c,d,e,f* prove that the model presented in this paper satisfies this criteria. Also the addition of agents should not disturb the actual operation of distance relaying scheme i.e. it should be deadlock free and be able to isolate faulted line within 1 sec. Properties *a,b*

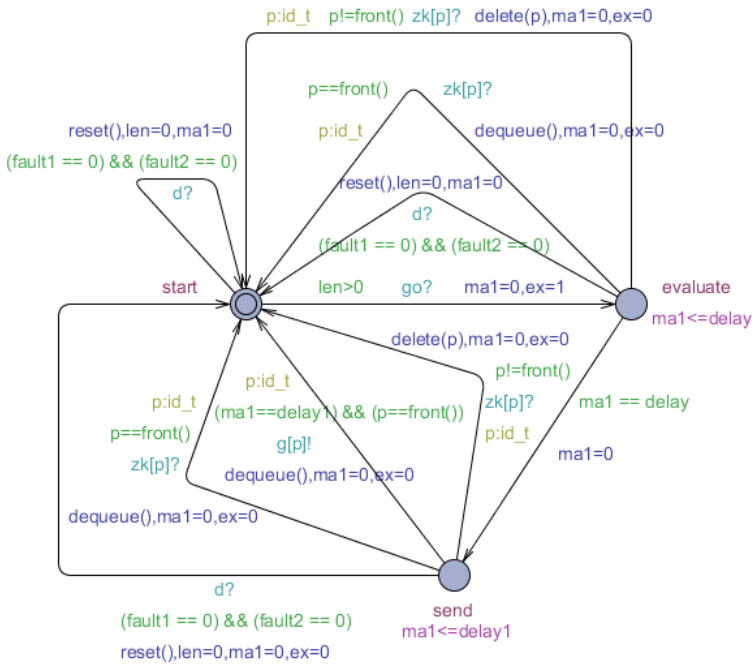


Fig. 12. Master agent task execute automaton

verify that these two requirements are met. Therefore the above described six logical properties are sufficient to guarantee the correctness of our model.

6 OBSERVATIONS

In the above two sections agent based Zone 3 relay supervision scheme is formally modelled and verified for the simplest scenario of a single transmission line being protected by two Zone 3 relays. Depending on the power system network topology, it is possible that more than two Zone 3 relays may be protecting a transmission line. The following observations discusses how to handle this scenario.

1. Observation 1: The number 'N' of Zone 3 slave agent requests a master agent with an average service time of t_s can handle at any given time is upper bounded by $N \leq (1000 - t_r)/(t_s)$. Where t_r is the maximum round trip communication delay between any slave agent and master agent in the network. It is possible that a Zone 3 slave agent may not receive acknowledgement from the master agent with in its fault clearing time of 1 sec. The two main reasons for this are network congestion and length of the queue at the master agent. In order to mitigate the network congestion problem, in OPNET simulations

we designed the network with sufficient bandwidth [2]. Therefore the problem of network congestion can be neglected. As mentioned earlier, from OPNET simulations the average t_s is assumed to be 133 msec and t_r is 150 msec which results in $N \leq 6.4$.
It is well known that the transmission line fault occurrence is a rare event. Further the probability of a fault occurring simultaneously in more than one transmission line is very low. Therefore we restrict this analysis to a single transmission line fault. Also it is mentioned earlier that we restrict our analysis to Zone 3 relay supervision scheme. As discussed above, with $t_s = 133msec$ the maximum number of slave agent queries answered by a master agent in 1 sec is 6. Table 2 shows the percentage of transmission lines in a given power system network protected by more than six Zone 3 relays. The percentage is around 18 for a 30 bus system and for remaining bus systems the percentage is less than 8. As the percentage of transmission lines with more than six Zone 3 relays is high, the master agent should be capable of handling more than 6 queries in a second. This can be achieved by doubling the query processing capacity of the server or arranging an extra server for query processing at the master agent. Either of these can result in the maximum number of slave agent queries answered by a master agent to be 12. It can be observed from Table 2 that the percentage of transmission lines in a given power system network protected by more than twelve Zone 3 relays is zero. Therefore for the power system networks shown in Table 2, a master agent capable of answering 12 queries per sec should be sufficient to meet the stringent timing requirements of Zone 3 relays. If the above discussed issues are taken into consideration, the formal models can be easily extended to a power system network of any size.

Table 2. Percentage of Zone 3 relays protecting a transmission line

Bus System	**% of lines with** $N > 6$	**% of lines with** $N > 12$
14	0	0
30	17.7	0
57	5.75	0
118	7	0.0025
127	3.25	0

2. Observation 2: It is aforementioned in section III that a larger bus system requires more than one master agent to answer queries from Zone 3 slave agents. Therefore a power system network is divided into sub-networks and a master agent is assigned to each sub-network to acknowledge queries from Zone 3 slave agents in that sub-network. It is possible that a network partitioned into sub-networks can be as shown in figure 13. If we can prove that both the sub-networks are disjoint, then the above described formal models and observation I can be applied to them to prove that both the sub-networks independently satisfy the properties verified in section VI. Therefore the entire power system network consisting of both these sub-networks can

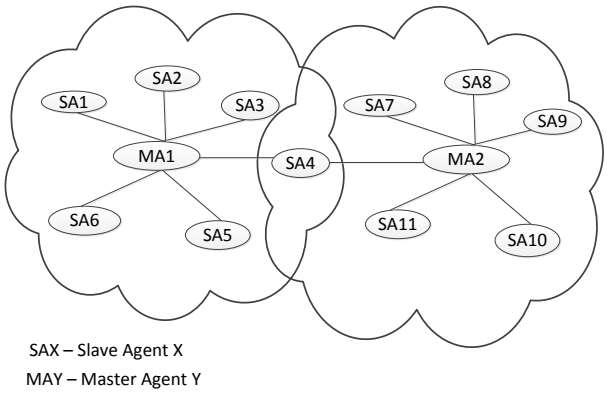


Fig. 13. Slave agent communication with multiple master agents

be assumed to satisfy the properties mentioned in section VI. The only connection between the two sub-networks shown in figure 13 is that there exists some slave agent relays that are considered as a part of both these sub-networks. If these relays sense a fault, they can send queries to the master agents in both the sub-networks and fault classification depends upon the response from both the master agents. Thus, there exists some interconnection between both the sub-networks. The interconnection can be avoided by using directional relays at the buses that are common to both the sub-networks. The directional relays can distinguish the fault i.e. in which sub-network the fault exists and based on that it can communicate with the corresponding master agent. Thus, the two sub-networks can be proved to be disjoint. If there are more than two sub-networks in a network, the same approach can be used to negotiate the interconnection between different sub-networks. As the sub-networks in a network are proved to be disjoint, each sub-network can satisfy the verification properties discussed in section VI and observation I. Therefore the entire network can satisfy the properties discussed in section VII.

7 Conclusion

In this paper we used a formal verification tool called UPPAAL to formally verify and validate agent based back up relay supervision scheme for transmission line protection system. Time based abstract formal models that capture the behaviour of sensors, breakers, relays, master and slave agent are described. The informal requirements of the agent supervised transmission line protection system are formalized in 6 logical properties and are verified and validated successfully. To the best of our knowledge this is a first attempt to use formal verification in power system protection. One of the future plans include modelling the probabilistic behaviours of the relays and find the reliability with which the Zone 3 relay

provides protection in the event of Zone 1 and Zone 2 failures. We plan to use PRISM model checker for this.

References

1. S. Garlapati, H. Lin, S. Sambamoorthy, S. Shukla, and J. Thorp. Agent based supervision of zone 3 relays to prevent hidden failure based tripping. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 256 –261, oct. 2010.
2. S. Garlapati and S. Shukla. Optimal location of master agents in an agent based zone 3 protection scheme designed for robustness against hidden failure induced trips. In *IEEE PES General Meeting, 2012*, July 2012.
3. S. Garlapati, S. Shukla, and J. Thorp. An algorithm for inferring master agent rules in an agent based robust zone 3 relay architecture. In *North American Power Symposium (NAPS), 2010*, pages 1 –5, sept. 2010.
4. S. Horowitz and A. Phadke. Third zone revisited. *Power Delivery, IEEE Transactions on*, 21(1):23 – 29, jan. 2006.
5. S. H. Horowitz and A. G. Phadke. Power system relaying. In *Research Studies Press Ltd.,*, 2004.
6. NERC. System Protection and Control Task Force. Report, Rationale for the Use of Local and Remote (Zone 3) Protective Relaying Backup Systems, February 2005. http://www.nerc.com/docs/pc/spctf/Zone3Final.pdf.
7. A. G. Phadke. Hidden failures in electric power systems. *International Journal of Critical Infrastructures*, 1(1):64–75, January 2004.
8. A. G. Phadke and J. S. Thorp. Computer relaying for power systems. In *second edition, Research Studies Press Ltd and John Wiley & Sons*, 2009.
9. T. S. *Analysis of power system disturbances due to Relay Hidden Failures*. PhD thesis, Virginia Tech, Dept of ECE, Blacksburg, VA, USA, 1994.
10. J. Thorp and A. Phadke. Protecting power systems in the post-restructuring era. *Computer Applications in Power, IEEE*, 12(1):33 –37, jan 1999.
11. H. Wang and J. Thorp. Optimal locations for protection system enhancement: a simulation of cascading outages. *Power Delivery, IEEE Transactions on*, 16(4):528 –533, oct 2001.