

Systematic Testing of Protocol Robustness: Case Studies on Mobile IP and MARS

Shamim Begum, Meeta Sharma, Ahmed Helmy, Sandeep Gupta
Electrical Engineering, University of Southern California
{sbegum, meetasha, helmy}@usc.edu, sandeep@poisson.usc.edu

Abstract

Systematic Testing of Robustness by Evaluation of Synthesized Scenarios STRESS is a methodology developed for the systematic testing of protocols, and includes algorithms for generating topologies and event sequences that rigorously test the correctness or performance of a given protocol. In this paper, we apply the STRESS method to Mobile IP (MIP) protocol and the MARS protocol for supporting IP-multicast over ATM. For each protocol, we develop a protocol model and analyze its robustness. We also analyze complexity of the STRESS test generation algorithms. In the process, we identify the limitations of the existing STRESS models and algorithms, and propose extensions to carry out our case studies.

With the aid of STRESS, we were able to identify several protocol behaviors that lead to error or performance degradation. For MIP we identified such behaviors with the crash of a home agent or the loss of a registration message. For MARS, undesired behavior was detected with the crash of MARS server or source and the selective loss of a join or leave message. The complexity of forward search was found to be $O(n^2)$ for both MIP and MARS. Incorporating the fault model in the search was found to affect the number of states searched. The crash of a home agent in MIP, and the crash of a data source or server in MARS, were both found to increase the number of states searched. However, the asymptotic complexity was not affected.

1 Introduction

Protocol verification is the process of ensuring the logical consistency of the protocol specification, independent of any particular implementation, and typically addresses *safety, liveness and responsiveness* properties. With the exponential growth of Internet in recent years and with growth of other services, the complexity of a typical network protocol has increased and along with it the protocol verification and testing has also become

more difficult.

Systematic Testing of Robustness by Evaluation of Systematic Scenarios (STRESS) [1] provides a framework for the systematic design and testing of protocols. STRESS uses a finite state machine (FSM) model of the protocol and uses a mix of forward and backward search techniques to generate the tests. The output tests include a set of protocol events and network failures that lead to violation of protocol correctness¹. In this paper, we extend the STRESS methodology and apply it to two protocols: Mobile IP (MIP) and MARS. MIP allows mobile hosts to send and receive packets addressed with their home address, regardless of their point of attachment to the Internet [6]. The main mechanism in MIP is the *registration* mechanism, through which the mobile node (MN) informs its home agent (HA) of its new location. Proper registration is essential for correct packet forwarding. If the registration mechanism is successful, then the home agent intercepts each packet for the mobile node and encapsulates and sends to the binding for that mobile node. At the other end, the *foreign agent* (mobility agent) decapsulates the packet and sends to the mobile node. If the registration mechanism is not correct, then this may lead to serious errors. It is because of this reason we study robustness of MIP, especially the registration mechanism.

The Multicast Address Resolution Server (MARS) provides mechanisms to support IP multicast services over ATM networks [11]. MARS acts as a registry of multicast group membership, storing the IP/ATM addresses of ATM endpoints who consider themselves to be members of a given IP multicast group at any instant of time - the set of such endpoints are defined as a *Cluster*. The server maintains a *point-to-multipoint* cluster control VC with all the cluster members on which it broadcasts group membership information. When a cluster member wants to join a group, it sends join request to the server that relays the join over cluster control Virtual Circuit (VC). An active source, upon reception of this relayed join message, adds the new receiver

¹The STRESS methodology is briefly described in Section 3.

as a leaf node of the VC connected to the group. Since the relayed join message is broadcast over cluster control VC, it might happen that the receiver which sent the join request received the relayed join while it gets lost from the server to the source - such a scenario leads to packet loss. We study the behavior of protocol under condition like selective loss of join and leave messages using STRESS algorithms.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 provides formalization of the problem. Details of the case studies are given in Sections 4, 5. Analysis and results are given in Section 6. Section 7 concludes.

2 Related Work

There has been significant research performed on verification of protocols, that address protocol safety and liveness properties. The two main approaches to this have been Theorem Proving and Reachability Analysis [9, 10]. In Theorem Proving, a set of axioms are defined and relations are constructed based on these axioms to prove the desirable properties mathematically. In Reachability Analysis algorithms, all the possible states are generated and all states reachable from a given state are inspected. However, this method suffers from the problem of *state explosion*. STRESS [1, 2] proposes a set of algorithms for generating test scenarios that target robustness and correctness violation, or worst case performance. STRESS has been applied to case studies on multicast routing protocols (PIM-DM and PIM-SM) and reliable multicast [5]. However, it has not been applied to mobility protocols or interoperability protocols. We investigate two protocols in those domains in our study. [15] presents a verification study for MIP, but it does not address protocol behavior in the presence of network faults. To the best of our knowledge, there has been no work done with regards to testing of MIP for robustness. Not much work has been done for studying MARS protocol. Issues have been identified that affects the cluster size a MARS server can support [12].

3 Overview of STRESS

STRESS [1, 2, 4] uses two approaches for test generation: 1. **Fault Oriented Test Generation (FOTG)** A fault oriented test is generated for a specific fault. It starts from the fault and synthesizes the necessary conditions to drive the protocol into error. This algorithm uses a mix of forward and backward searches [2].

2. **Fault Independent Test Generation (FITG)**

Fault Independent TG works without targeting individual faults as defined by the fault model. Such an approach may employ a forward search technique to inspect the protocol state space after integrating the fault into the protocol model [4].

To use the STRESS methodology, we need to specify the system model, i.e., protocol and message semantics in a processable form, and the correctness criteria. The model is then processed by the test generation algorithms that use search techniques to synthesize the test scenarios. The result is a set of network topologies, protocol event sequences, network failures, that violate the correctness criteria.

The finite state machine (FSM) formalism is used to represent the protocol. The interaction among the entities of the protocol is modeled by a global FSM (GFSM) [1]. The system consists of network and topology elements and a fault model. A test input pattern can be defined as a 3-tuple [*Topology, Events, Faults*], where Events is a sequence of host events [1, 2, 4], and faults are low level anomalous behaviors that may affect the protocol under test. Faults include *loss of packets due to congestion, loss of state and delays*. The fault model is integrated into the global FSM. In our case studies we use single selective message loss and machine crashes as fault models.

1. **FSM Model:** Every instance of the protocol is modeled by a deterministic FSM consisting of (i) a set of states (ii) a set of stimuli causing state transition, and (iii) a state transition function or table describing the state transition rules. For a system i , this is represented by the machine $M_i = (S, \tau_i, \delta_i)$, where S is a finite set of state symbols, τ_i is the set of stimuli, and δ_i is the state transition function.

2. **Global FSM Model:** The global state is defined as the composition of states of the individual entities of the system. The output messages from one entity may become input messages to other entities. Such interaction is captured by the GFSM model in the global transition table. The behavior of a system with n entities may be described by $M_g = (S_g, \tau_g, \delta_g)$, where $S_g : S_1 \times S_2 \times \dots \times S_n$ is the global state space, $\tau_g : \cup_i^n \tau_i$ is the set of stimuli, and δ_g is the global state transition function $S_g \times \tau_g \rightarrow S_g$.

We use a combination of the test generation methods to analyse the behavior of these two protocols in error condition and to study the complexity of the protocol state space. For each of the protocols, we identify the correctness criteria, states representing different entities of the protocol, messages that affect these states and design transition tables that describe the protocol mechanisms. For the message to be lost, we derive the minimum topology G required to trigger the mes-

sage using FOTG algorithm [2]. We then apply forward search to G to study the recovery processes and backward searches to G to study its reachability. We also apply forward searches while increasing the network size and analyze the complexity of the protocol state space. Similarly, we apply backward searches while incorporating the faults. Mobility introduces the problem of dynamics and handoff. We also study more complex multicast protocols, like MARS, for the inter-operability using multicast, which makes the problem more complex.

4 Case Study: Mobile IP

Mobile hosts are nodes that dynamically change their point of connectivity to the Internet. Mobile IP (MIP) defines a mechanism which enables nodes to change their point of attachment without changing their IP address. MIP [6] allows mobile hosts to send and receive packets addressed with their home network IP address, regardless of their current point of connectivity. The challenge for supporting mobility at the IP layer is handling address changes. The mobility of the host is hidden through the use of home and foreign agents that handles the routing of packets to the mobile host. A mobile node discovers its connectivity to the network via the agent discovery mechanism. If it has moved, it tries to register with a foreign agent through registration mechanism [6].

The home agent intercepts datagrams destined for the mobile and tunnels them towards the new care-of-address of the mobile at a foreign network.

According to MIP, we can define a *Mobile Node* as a host or a router that changes its point of attachment from one network to another network. A *Home Agent* is a router on a mobile node's home network which tunnels datagrams for delivery to the mobile node when it is away from home, and maintains current location information for the mobile node. A *Foreign Agent* is a router on the mobile node's visited network which provides routing services to the mobile node while registered. The foreign agent de-tunnels and delivers to the mobile node. For datagrams sent by the registered mobile node the foreign agent may act as a default router. In general, home agents and foreign agents are termed as *Mobility Agents*. A *Care-of-address* is simply the end-point of the tunnel, i.e., the tunnel terminates at the care-of-address of the mobile node. It should be an address to which the datagrams can be delivered via conventional IP routing. At the care-of-address the original datagram is removed from the tunnel and delivered to the mobile node.

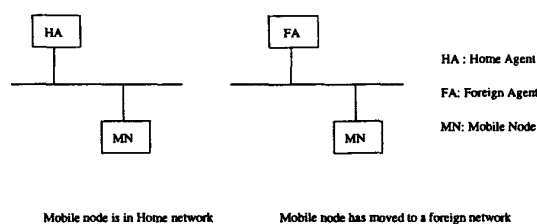


Figure 1: Topology considered for MIP study

4.1 MIP Mechanisms

Mobility agents (home/foreign agents) advertise their presence by broadcasting agent advertisements. A mobile node may optionally solicit an agent advertisement message from any locally attached mobility agents through an agent solicitation message. A mobile node receives these agent advertisements and detects whether it is on home network or on foreign network. If the mobile node detects that it has moved from its home network, it obtains a care-of-address on the foreign network. If the mobile node detects that it is still in its home network then it functions without any mobility services. If it detects that it is returning to its home network from being registered elsewhere, the mobile node de-registers with its home agent through exchange of a Registration Request and Registration Reply message with it.

The mobile agent operating away from home then registers its care-of-address with the home agent through exchange of registration request and registration reply message. Datagrams sent to the mobile node's home address are intercepted by its home agent, tunneled by the home agent to the mobile node's care-of-address and then received at the tunnel end-point and delivered to the mobile-node. Datagrams sent by the mobile-node are delivered to their destination using standard routing mechanisms.

4.2 Modeling the Protocol

Figure 1 shows the topology considered. In our model we consider a LAN with a home agent and another LAN with a foreign agent².

As described in Section 3, a protocol is modeled as FSM and the interaction between the different entities of the protocol are captured by the Global FSM. The modeling of the mobile node, Home Agent and Foreign

²We present the one FA model for simplicity and illustration. Our case study actually considers multiple FAs, as shown in Section 6.

State Symbol	Meaning
InHA _i	MN _i in it's Home Network
DA _i	MN _i discovering an Agent
Reg _{mn_i}	MN _i tries to register its new care-of-address
DeReg _{mn_i}	MN _i deregisters the binding _j
InFA _{ij}	MN _i registered with FA _j
EM _i	MN _i has lost it's state
NMode _i	HA has no binding for MN _i
T _{ij}	HA _i encapsulates the packets to the FA _j
EH _i	HA _i has crashed.
NMN _i	FA _j has no binding for MN _i
Reg _{fa_i}	FA _j tries to register MN _i with the home agent
DeReg _{fa_i}	FA _j deregisters the binding for MN _i
DT _{ij}	FA _j decapsulates and sends the packet to the MN _i

Table 1: System states for MIP

Timer	Description
RegLifetime _{ij}	Life Time of the registration
RegReq _{ij}	MN _i /FA _j send a Reg request again if the timer expires before getting a reply
DeRegReq _{ij}	MN _i sends a DeReg request again if the timer expires before getting a reply

Table 2: MIP timer table

Agent is performed according to the protocol mechanism described in Section 4.1.

A finite state machine consists of a set of states, set of stimuli which cause the state transitions and a state transition table which describes the state transition rules. In our system we have considered a LAN with one home agent, one or more foreign agents and one Mobile node.

4.2.1 FSM Model

For a specific Mobile Node (MN)-Foreign Agent (FA) pair, we define the states w.r.t a specific LAN to which the Foreign Agent (FA_j) is connected.

All possible system states are listed in Table 1. The various timers modeled are listed in Table 2.

The possible states for mobile node, home agent, foreign agent are as follows:

- For Mobile Node: InHA_i, InFA_{ij}, Reg_{mn_i}, DeReg_{mn_i}, EM_i
- For Home Agent: NMode_i, T_{ij}, EH_i
- For Foreign Agent: NMN_j, DT_{ij}, Reg_{fa_i}, DeReg_{fa_i}

Stimulus	Description
RegReq _{mn}	Request for registration sent from MN to FA
RegReq _{fa}	Request for registration sent from FA to HA
RegRep _{grt_{ha}}	Registration granted by HA to FA
RegRep _{rej_{ha}}	Registration rejected by HA to FA
RegRep _{grt_{fa}}	Registration granted by FA to MN
RegRep _{rej_{fa}}	Registration rejected by FA to MN

Table 3: MIP protocol messages for Registration mechanism

Table 3 lists the stimuli for the registration mechanism.

Global FSM Model The global state is given in the order of home agent, mobile node, foreign agents. An example global state for a case with two foreign agents, with mobile node registered with FA_j, is given by T_{i1}, InFA_{ij}, DT_{ij}, NMN_k.

4.2.2 Transition Table

The transition table describes the condition of occurrence of each stimulus. A condition is given as stimulus, and state or transition. At least one pre-condition is required to trigger the stimulus. A post condition is a stimulus or transition which is triggered by the stimulus. Table 4 is the transition table for the registration mechanism explained in Section 4.1.

4.2.3 Fault Models

In our study we consider a *single fault model*, i.e., only one fault can occur at a time. The various fault models incorporated in the study are as follows: (1) Packet Loss: We consider the control message loss which can lead to further loss of packets. (2) Loss of State: The HA/MN/FA agent can loose the state information, such as the information about the present binding of the mobile node. (3) Crash: We consider crashes from which there is no recovery.

4.2.4 Protocol Errors

Protocol errors can be defined in terms of end-to-end behavior as functional correctness requirements. Some of the protocol errors identified for MIP are:

1. Blackholes: When a mobile node moves into a new network region, there is an exchange of packets called a handoff and during this period, all normal transmissions are disrupted. This can lead to data loss, and sustained loss of data packets can lead to blackholes.
2. Duplication: Multiple copies of the same packet is received by the mobile node.

Stimulus	Pre-condition	Post Condition
RegReq _{mn}	AgentAd _{fa} .(DA → Reg _{mn})	ReqReq _{fa} .(NMN → Reg _{fa})
RegReq _{fa}	RegReq _{mn} .(NMN → Reg _{fa})	RegRep _{grt_{ha}} .(NMode → T), RegRep _{rej_{ha}}
ReqRep _{grt_{ha}}	RegReq _{fa} .(NMode → T)	RegRep _{grt_{fa}} .(Reg _{fa} → DT), RegRep _{rej_{fa}} .(Reg _{fa} → NMN)
ReqRep _{rej_{ha}}	RegReq _{fa} .(NMode → T)	RegRep _{rej_{fa}} .(Reg _{fa} → NMN)
ReqRep _{grt_{fa}}	RegRep _{grt_{ha}} .(Reg _{fa} → DT)	RegLife.(Reg _{mn} → InFA)
ReqRep _{rej_{fa}}	RegRep _{grt_{ha}} .(Reg _{fa} → NMN) RegRep _{rej_{ha}} .(Reg _{fa} → NMN)	AgentSol.(Reg _{mn} → DA)

Table 4: Transition table for MIP

3. Registration Latency: There would be no packet delivery until the registration mechanism is successfully completed.

4. DeRegistration Latency: While a mobile node is trying to deregister, packets are forwarded to the foreign agent until the deregistration is granted. The foreign agent remains a part of the forwarding tunnel until the deregistration has been granted.

4.2.5 Correctness Criteria

The correctness criteria identified for Mobile IP are:

1. For each LAN, there must be exactly one home agent for a LAN. If there is no home agent, then there would be no packet forwarding to a mobile node in a foreign network.
2. If a move has been detected by a mobile node, then there *must* exist a mobility agent to serve that mobile node. The absence of a mobility agent would lead to a *black-hole* problem.
3. For every mobile node i and home agent j , if j is i 's home agent and i 's current registration is valid and has not expired, then the mobility binding of i in its home agent's binding table is valid and has not expired.
4. Consistency of registration: The mobility agent with which the mobile node i is believed to be registered (by the home agent) should actually be the one for which i is currently registered.

If the mobile node has state FA_{ij}, i.e., it MN_i is in FA_j, then the HA should be in T_{ij} and FA should be in state DT_{ij}. States like (T_{ij}, InFA_{ik}, DT_{ij}) and (T_{ik}, InFA_{ij}, DT_{ij}) are incorrect.

4.3 MIP Study Results

In this Section, we give a brief description of some of the error scenarios which can lead to some correctness violation or degradation in performance. For each study we describe the forward and backward implication. We have based our study on the protocol version in [6], and so our model presents a study of the basic protocol mechanisms and possible errors. Furthermore, the

model of the protocol that we used to generate the scenarios did not include periodic timers. However, the generated scenarios presented here have all been validated for a complete model of the protocol, including periodic timers.

4.3.1 Loss of Register Grant message with one Foreign Agent

1. **Topology Synthesis:** We start with the message RegRep_{grt_{fa}} and the FOTG algorithm generates the topology necessary for the message to be triggered. This is done iteratively considering all the preconditions and corresponding stimuli of preconditions. The topology synthesized is represented as G_i and is given by:

$G_i = N_{node,i}, Reg_{mn,i,j}, Reg_{fa,i,j}$. This represents the state of GFSM in which the RegRep_{grt_{fa}} can be applied.

2. **Forward Implication:** We then apply RegRep_{grt_{fa}} to G_i in cases of losses and no losses. In this process we note the state of the GFSM considering no loss of RegRep from foreign agent to the mobile node. G_{i+NoLoss} and G_{i+Loss} represent the state of the GFSM for them respectively.

$$G_{i+NoLoss} = T_{ij}, InFA_{ij}, DT_{ij}$$

$G_{i+Loss} = T_{ij}, Reg_{mn,i,j}, DT_{ij}$ The mobile node would keep rejecting the packets it gets and there would be an additional loss of packets. This state would persist until the RegRequest timer expires and the mobile node again tries to register.

The expected recovery time for this error is RegRequestRetx Timer + Time Taken in processing the registration message.

Loss of register grant message leads to blackholes. There would be a degradation in the performance of the system, as all the packets are lost until there is a successful registration.

4.3.2 Loss of Register Grant message with more than one Foreign Agent

If there exist more than one foreign agents on the LAN, the mobile node would get agent advertisements from

multiple foreign agents. The MN attempts to register with one of them, say FA_j . Here, we describe a simple case with two foreign agents (FA_j, FA_k) on the same LAN. We assume that the mobile node is trying to register with FA_j .

1. **Topology Synthesis:** $G_i = N_{mode_i}, Reg_{mn,i}, Reg_{fa,i}, NMN_k$

2. **Forward Implication:**

$$G_{i+N_{oLoss}} = T_{ij}, InFA_{ij}, DT_{ij}, NMN_k$$

The mobile node would keep rejecting the packets it gets and there would be an additional loss of packets. The mobile node would go into a discover agent state and would seek another agent. If the mobile node gets an advertisement from FA_k , it would then try to register with that agent.

$G_{i+Loss} = T_{ij,ik}, InFA_{ik}, DT_{ij}, DT_{ik}$, which is an incorrect state for there is inconsistency at the mobile node about its bindings. We assume that the home agent allows multiple bindings for the same mobile node. Otherwise, the severity of this error is higher.

4.3.3 Crash of a Home Agent

If the home agent crashes then the mobile node and foreign agent may be in the $InFA_{ij}$ and DT_{ij} states respectively expecting packets from the home agent, whereas there is no home agent.

1. **Topology Synthesis:** $G_i = T_i, InFA_i, DT_i$

2. **Forward Implication:**

$$G_{i+crash} = EH, InFA_i, DT_i,$$

This is an incorrect state as there is no home agent for the system. This violates correctness criteria. We assume a *hard crash* for our system. There is no recovery from this state, as when the home agent crashes, the mobile node has no way of getting the packets destined for it, and it can never detect its point of connectivity properly³.

4.3.4 Loss of State for the Mobile Node

When the mobile node loses its state, it does not know the address of the home agent⁴ unless it gets an advertisement from the home agent. If the mobile node has moved to another foreign network, it would not be able to register with the foreign agent, as it does not have any information about its home agent. This can be recovered from when it returns to its home network and gets an advertisement from the home agent.

³We can see that a design that allows only one home agent without an election mechanism to elect an alternate one in case of failure, is a poor system that is susceptible to single-point of failure errors

⁴It is also possible for a mobile node to use option 68 specified in [6] for DHCP to get a home address and a home agent address, but this approach is not taken by this protocol

The recovery in this case depends on where the mobile node is when it loses state. If the mobile node is in the home network, then it would recover as soon as it would receive a advertisement from its home agent. But if it is in a foreign network when it loses state, it would not be able to register, as it would not have any information about the home agent. It can only recover when it returns to its home network.

5 Case Study: MARS

In this section, we evaluate the utility of STRESS by applying it to Multicast Address Resolution Server (MARS) protocol. Multicasting over ATM network is achieved in two models - Virtual Circuit (VC) mesh based and Multicast Server (MCS) based. In both of the models, the MARS maintains the registry of multicast group membership by storing the IP/ATM addresses of ATM hosts and thus maintains the cluster. The first model gives the most fundamental approach of multicasting where each source establishes its own point to multipoint VC with group members analogous to source based tree. In the MCS based model, all sources establish a point to point VC with an intermediate server called Multicast Server (MCS) that forwards the data to the group members analogous to shared tree [11]. For our case study, we take the mesh based approach.

5.1 Protocol Mechanism

In order to participate in multicast on ATM networks, all ATM hosts need to register with the MARS server. Once registered, the cluster members keep receiving updates on group membership information that is broadcast on the cluster control VC by the server. The functions of multicast data delivery can be described in terms of the following mechanisms.

When a host wants to be a member of the cluster, it sends a Registration message to the server including its ATM and IP address. Upon reception of this message, the server adds the host as a leaf node of the cluster control VC and sends its unique cluster member identifier. The server stores the address mapping for all of the registered cluster members. When a member wants to deregister, it sends Deregistration message to the server. Upon reception, the server drops it off the cluster control VC.

When a cluster member wants to join group it sends Join message to the server and starts retransmission timer. Server relays the Join over cluster control VC. Upon reception of this relayed Join from server, the host which sent the join turns its timer off. Upon reception of relayed Join, a source sending to the group adds the

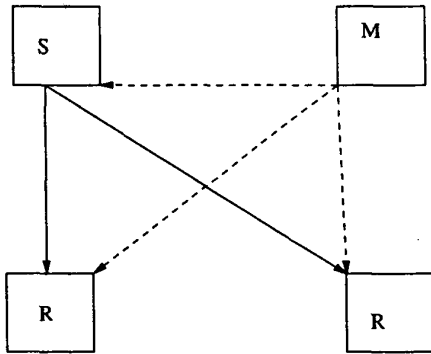


Figure 2: MARS VC-mesh based model

new receiver as a leaf node of the point-to-multipoint VC with the group members. In case of receiver leaving, the source drops the leaf off the VC.

When a host wants to send to a group, it sends a Request to the server for the address mapping of the group. The server replies back with list of IP and ATM addresses of the group members. The source then establishes a point-to-multipoint VC of which a receiver becomes a leaf node. Each source also maintains a revalidation timer associated with that VC for the group. When expires, the timer turns a flag indicating the source to revalidate membership with the server. Consequently, when the active source gets packet to send, it starts revalidating the group membership information if the revalidation flag for the VC is set. The revalidation starts process by sending Request to the server. This process does not disrupt the source from sending packet to the group. In contrast, if the revalidation flag is not set, then the source directly sends the packet to the group without initiating revalidation.

5.2 Modeling the Protocol

Figure 2 shows the VC-mesh based model we consider for our study where there is one MARS server serving the cluster. There are three entities in the protocols: server, source and receiver. As in multicast, sources need not be a members of the group to send to the group. A source can send to multiple groups, however, in our model we only consider source to send to only group and receivers to join that group. Our initial model only considers single source and we study the behavior under conditions where there can be multiple sources sending to the group.

State Symbol	Meaning
M_i	M_i is serving as MARS
M_{iRel}	M_i is relaying Join/Leave over cluster control VC
ES_i/ER_i	S/R has no state information
CWS_i/CWR_i	End point i joined cluster and willing to be S/R to the group
CS_i	S_i has no mapping for the group
S_i	End point i has point-to-multipoint VC setup with the group
$S_{iUpdate}$	S_i is updating its point-to-multipoint VC with group
PS_i	S_i is sending to the group
CR_i	R_i has received Join relayed by server
NR_i	R_i has been added as leaf node to the point-to-multipoint VC rooted at S
R_i	R_i is receiving multicast data from S

Table 5: System states for MARS

Timer	Description	Values
Retx	Join/Leave/Registration Retransmission interval	10 secs
Rev	Revalidation interval	10 secs

Table 6: MARS timer table

5.2.1 FSM Model

We define states of MARS server M , data source S and receiver R according to the protocol mechanisms. A subset of possible system states are listed in Table 5. The various timers modeled are listed in Table 6 - the values specified here are as recommended.

The possible states for server, source and receivers are as follows:

- For MARS server: M , M_{Rel} , M_{Rep}
- For Source: ES , CWS , CS , S , S_{Update} , PS
- For Receiver: ER , CWR , CR , NR , R

Stimulus	Description
Req	request from sender/client to MARS for group membership
Rep	MARS Reply to the sender/client with group membership mapping
Join	Join request from a client to MARS
JRelay	Join message when relayed by MARS over cluster control VC
Reg	Registration request from a non member node to MARS
Update	Setup unicast/multicast VC add/drop a node from VC etc

Table 7: MARS protocol messages for Join mechanism

Global FSM Model: GFSM represents the combined states of all entities of the network. An example global state representing a network with server, one source and two receivers is: M_i, PS_j, R_k, CWR_l . In this network, i is the server, receiver k is receiving multicast data from source j and host l has joined cluster and willing to join the group. In this case study, we present GFSM in the order of server, source and receivers to present the topology or network.

5.2.2 Transition Table

Table 8 and Table 7 show a subset of the transition table and protocol messages respectively, for our model described in the previous section. The transition table describes the Join mechanism. The precondition of a stimulus is the condition required for the stimulus to be triggered - for example, for a host to trigger a Join, it must receive a HJoin from the higher layer. The post-conditions are the set of stimuli and transitions that this stimulus triggers - for example, a Join causes the state of server to change so that it triggers JRelay. Similarly, JRelay causes the source to trigger Update. In this model, Update represents all signaling functions available to local AAL user of the end points UNI interface, e.g., Establish unicast/multicast VC, add new leaf node to a previously established VC etc. In our model, the Update message is always deterministic which implies that a source always is able to add a new leaf node to the VC with the group to which it is sending or drop an existing leaf node from that VC.

5.2.3 Fault Model

In our case study, we consider a single-fault model where only one fault can occur at a time. We define the following as the fault model:

1. Packet loss due to congestion, VC failure, insufficiency of resources required for assembly and reassembly at AAL5. Therefore, packets are delivered correctly or dropped before arriving at the destination.
2. Loss of state, such as address translation information and cluster membership information due to insufficient memory, momentary crashes, etc.

5.2.4 Protocol Errors

For MARS and other multicast protocols, an error can occur in one of the following ways:

1. Join latency: The time required by a receiver joining the group to start receiving packet sent to the group.
2. Leave latency: The time required by a receiver leaving the group to stop getting packets sent to the group.

It leads to unnecessary packet reception after a receiver leaves the group.

3. Packet duplication: Multiple copies of a packet are received by a receiver.
4. Blackholes: Consecutive packet loss between periods of packet delivery.
5. Registration latency: Lack of cluster membership information delivery after a host joins a cluster.
6. Deregistration latency: Unnecessary delivery of cluster membership information after a host leaves a cluster.
7. Revalidation latency: Lack of packet delivery when a source revalidates the group membership information, connects or disconnects a leafnode. This in turn leads to blackholes.

5.2.5 Correctness Criteria

Based on the fault model, we define the correctness criteria that we assume to be provided by the protocol designer. Once these conditions are satisfied, the state machine is assumed to be correct. Violation of these conditions in a stable state leads the state machine into error.

1. A host must be a registered cluster member to send to a group and to receive from a group. Violation of this condition may lead to packet loss, i.e., registration latency, registration and join latency or blackholes. For example, GFSM states like (M_i, CWS_j, R_k) are not correct.
2. When a host joins a group with an active source, it must be added as a leaf node of the point to multipoint VC rooted at that source for the group, i.e. the source must update its VC upon its join. Violation of this condition leads to revalidation latency. The revalidation latency is defined as the time required for a source to request group membership information, receive replies from server and add a new leaf node. For example, GFSM states like (M_i, CS_j, CR_k) , (M_i, S_j, R_k, CR_l) are incorrect.
3. When a source receives group membership information from the server for a group with nonzero members in that cluster, the source must have point to multipoint VC rooted at itself. Violation of this condition leads to blackholes.
4. When a source sets up point to multipoint VCs for a group, the number of leaf nodes of the VC must be equal to the number of group members in the cluster. Violation of this condition may lead to join latency or black holes. States like (M_i, S_j, ER_k) are incorrect.
5. When a receiver is receiving packet, there must be a source that has point-to-multipoint VC through which the receiver is receiving the packet. Violation of this may lead to registration and revalidation latency. States like (M_i, ES_j, R_k) are incorrect.

Stimulus	Pre Condition	Post Condition
Join	HJoin.(CWR → CWR _{TRetz})	(M → M _{Rel}).JRelay
JRelay	Join.(M → M _{Rel})	S → S _{Update} .Update
Update	JRelay.(S → S _{Update})	S _{Update} → S, CWR _{Retz} → CR

Table 8: Transition table for MARS: Join mechanism

5.3 MARS Study Results

We apply FOTG to study the robustness of the protocol in the presence of message loss and loss of state. We present the scenario that violates the correctness condition and hence leads to an error state. For each of the study, we describe the steps for topology synthesis, forward implication and backward implication.

5.3.1 Selective Loss of JRelay

JRelay is broadcast by the server over cluster control VC. Here we consider the loss of the message from server to the source. We study this scenario for two cases: for loss of JRelay i) when the first receiver joins and ii) when the n^{th} receiver joins, since the state machine recovers from the error in different ways in these two cases.

Join of the First Receiver

1. **Topology Synthesis:** We start with the message JRelay and the method generates the topology necessary for it to be triggered. This is done by iteratively considering all the preconditions and corresponding stimuli of preconditions. The topology thus synthesized is represented as G_i and given by: $G_i = M_{iRel}, CS_j, CWR_{kRetz}$. This represents the state of GFSM in which JRelay can be applied.

2. **Forward Implication and Recovery:** We then apply JRelay to G_i . In this process we note the state of GFSM considering selective loss of JRelay from server to the source. $G_{i+NoLoss}$ and G_{i+Loss} represent the state of GFSM considering no loss of JRelay and selective loss respectively. From our study $G_{i+NoLoss} = M_i, S_{jUpdate}, CR_k$ and $G_{i+Loss} = M_i, CS_j, CR_k$ where G_{i+Loss} is a stable state that violates correctness criteria 2 - receiver k assumes to join the group, but source does not add any VC. Since there is no VC for the group, the revalidation process does not occur at this state, however, whenever the source gets a packet to send, it sends request to server and gets the address mapping of the newly joined receiver. In this case, there was no VC setup for the group, in fact, this loss of join message does not lead to any packet loss. The recovery time of the state machine is defined by the

time to process Request, Reply and Update messages.

Join of the n^{th} Receiver

1. Topology Synthesis for $n = 2$:

$$G_i = M_{iRel}, S_j, R_l, CWR_{kRetz}$$

2. **Forward Implication and Recovery:** $G_{i+NoLoss} = M_i, S_{jUpdate}, R_l, CR_k$, and $G_{i+Loss} = M_i, S_j, R_l, CR_k$, this is a stable state that violates correctness criteria 2. The state machine comes out of this error because of revalidation process. When the revalidation timer expires, it changes the state and when the source gets next packet to send, it starts the revalidation process by initiating a Request to the server. However, it does not stop the source to send data packet over the VC for the group, so it leads to packet loss for a period of time defined as follows.

Recovery Time (Revalidation Latency) = Revalidation Timer interval + Time to process Request, Reply and Update messages.

Backward implication: Using backward implication the algorithm generates the sequence of messages sufficient to trigger JRelay message. The messages are SPkt, RegRep, HReg for the source and HJoin, HReg for the receiver.

5.3.2 Selective Loss of LRelay

We apply FOTG for this message - for brevity we only present the results we have obtained from this study. Like JRelay, LRelay is broadcast over the cluster control VC. The state machine comes out of the error state because the revalidation timer expires and same sequence of events occur independent of the number of receivers in the network. This shows that the receiver which has already sent Leave message to the server, continues to receive packet from the source for the Recovery time period defined in the previous subsection. This leads to unintended packet reception during leave.

5.3.3 Crash of a Source

When a source crashes it momentarily loses its state and is triggered to empty state. This kind of crash can occur at any time and hence, we synthesize the topology required for all consecutive messages that may lead a

cluster member to a source. Then we apply forward implication to study the behavior of systems states after crashes. First, we apply SCrash and arrive at the state of the system as (M, ES, R). This is an error state that violates condition 5. The behavior of the system after application of host stimuli are as follows.

1. SPkt after SCrash: Since the source loses its state, this leads to registration and revalidation latency before it recovers from the error state. The recovery time is defined as: Revalidation Time + Registration latency + Time to process Request, Reply and setting up VC with the receivers. During this period, no receiver will receive any packet from the source.

2. HLeave after SCrash: We apply this stimulus to the state (M, ES, R). In this state, when the receiver sends Leave to server and the server multicast RRelay, the source does not have enough state to respond to this message. In this case, the system did not recover and it has to wait until the source gets a packet to send to the group and starts registration and revalidation.

5.3.4 Crash of a Receiver

The same steps are followed in this case. For simplicity we only present our findings of forward implication after crash. When the leaf node of the source corresponding to the receiver drops, the source starts revalidating the group as soon as it gets packet to send. However, the server also disconnects the leaf node associated with the receiver. Thus, the source does not add the receiver as its leaf node. This leads to blackhole in the receiver in case of SPkt after receiver's crash. If HJoin is applied after the crash, the receiver triggers registration and join that leads to the registration and join latency and eventually recover from the error.

1) System with only one receiver: One possible minimum topology for a network with only one receiver (that crashes) is (M, S, ER) - this is an error state that violates condition 4. The system recovery process depends on the sequence of events that might happen after the crash. If the source sends a packet before the revalidation timer expires, the receiver still gets packet from the VC - there is, in fact, no loss due to crash of the receiver. If the revalidation timer expires before the source has a packet to send, the source starts revalidation process during which the source keeps on sending packet over the cluster control VC and the receiver in turn does not lose any packet.

2) System with more than one receiver: In this case, one possible minimum topology is (M, S, ER, R) that satisfies all correctness criteria. The system is not in error. However, when the receiver comes up it leads to registration and join latency.

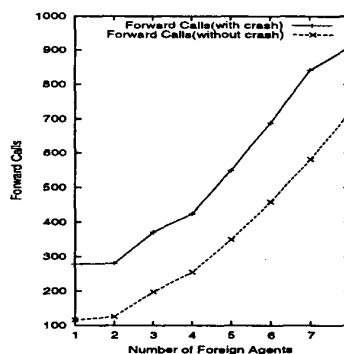


Figure 3: MIP: Number of forward calls vs number of Foreign Agents

5.3.5 Crash of the Server

If the server crashes and there is no election mechanism, the on-going activities of the group remains unaffected. For example, the source keeps sending to the group. However, when the revalidation timer expires or when a new host wants to join cluster or group, they send group address request or registration/join request to server which does not have any state information about the group or the cluster. This leads to an error from which there is no recovery. If there are multiple servers in the network, the scenario would be different because of the server election mechanism and interaction between the servers.

6 Complexity Analysis of the Search Algorithms

We applied FITG to study the complexity of the state space. The search results show the behavior of the algorithm with and without including faults. We identified the initial states for the network and applied forward search to it. We present the results for both of the case studies in this section.

The number of expanded states represents the number of visited stable states. The number of forwards represents the number of times the state machine was advanced forward denoting the number of transitions between stable states. The number of error states represents the number of expanded states that violate the correctness condition.

We study MIP for the complexity of FITG algorithm by varying the number of foreign agents. Only one home agent and one Mobile node were considered. Figures 3, 4 shows the comparison for the complexity with respect

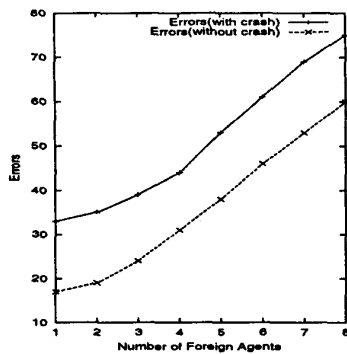


Figure 4: MIP: Number of errors vs number of Foreign Agents

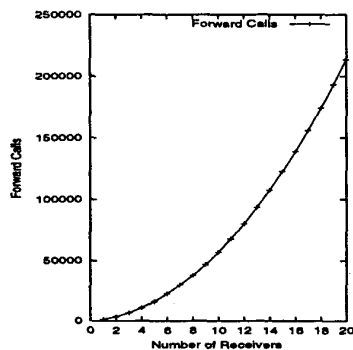


Figure 5: MARS: Number of forward calls vs number of receivers

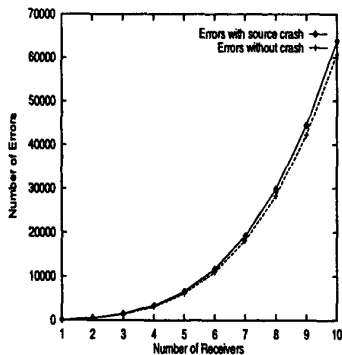


Figure 6: MARS: Number of errors vs number of receivers

to the number of forward calls and the number of errors. The model was analyzed for complexity for cases with and without crash of home agent. The crash was a *hard* crash (from which there is no recovery). Crash of the home agent leads the system into an unrecoverable error state. The mobile node cannot receive any packets if it is not in the home network. Due to the error being unrecoverable, the complexity of the search increases and so do the errors. The forward search complexity is $O(n^2)$, where n is the number of foreign agents.

We study the complexity of FITG algorithm for MARS protocol by varying the number of receivers as shown in Figures 5 and 6. The number of expand and forward calls increases non-linearly with number of receivers. The number of calls are $O(n^2)$ where n is the number of receivers in the network. The number of error states also increases non-linearly with number of receivers. We study the algorithms by varying number of sources and our study shows that the complexity is independent of the number of sources. The reason is that a source only needs to register to the server as cluster member to send to the group. Once it is added as a cluster member, it does not add any complexity as long as there is no change in the group. We also study the complexity of search and errors with the crash of the source. We found that the complexity as well as errors increase when the source crashes, though not significantly. The reason is that the crash is a soft crash, not a hard crash⁵ - the source eventually recovers from the error and the recovery process leads to Registration latency. When the source crashes, it goes to an empty state that is also its initial state. In this state, depending on the number and states of the receivers, it might be in error or in correct state. For example, if at least one receiver is in receiving state, it becomes an error state, while on the other hand, if all receivers are in states prior to joining the group, it becomes a correct state. In case of an error, the source starts registration and comes out of the error state eventually. The complexity of FOTG algorithm increases non-linearly with the increase of number of receivers.

7 Summary and Future Work

We analyzed MIP and MARS to study their correctness and robustness. Using the STRESS method we were able to detect protocol errors, generate the exact sequence of events which cause each of these errors and define the time that is required for it to recover from the errors. In both cases, the users of the protocol would loose data packets for reasonably significant periods of

⁵In case of server crash, which is a hard crash, there is an abrupt increase in complexity of search algorithms

time. For MIP we have studied the behavior for loss of control messages and also crash of home agent. A non-deterministic behavior in the registration mechanism was observed. In our model, we have not modeled the method of rejecting or granting the registration request, i.e., the different issues the mobility agent would consider before accepting or rejecting the request. The complexity of the search increases as $O(n^2)$, where n is the number of foreign agents. However, the complexity increases with the introduction of crash of the home agent. For MARS, we have studied the behavior for selective loss of control messages and crashes of source, receiver and server. Our results show that, in general, the complexity increases as $O(n^2)$, where n is the number of receiver and remains same with the increase of number of sources. However, crashing of source increases the complexity and errors slightly because of the error due to crash being recoverable.

Both of the case studies are based on the specifications [6, 11], which only describe the basic mechanisms. The modeling done for MIP does not include the authentication mechanisms. We need to add mechanisms which decide whether a registration has to be granted or rejected. Enhancement to the basic mobile IP, like route optimization, caching at the foreign agents, have not been considered in the present model. Though including these mechanisms would increase the state space, it would capture the interactions with the correspondent hosts and also between the foreign agents. The STRESS methodology does not include the support for periodic timers. The method must also be extended to incorporate the notion of real-time.

In MARS model, adding a leaf node to a point-to-multipoint VC was deterministic. In order to be non-deterministic, we need to extend the model to incorporate the lower layer mechanisms. For example, we need to add features in our model so that any source has the ability to decide whether a new leaf node can be added or not based on lower layer mechanisms. Also, we did not study the protocol behavior when there are multiple servers in the network because our model was based on [11], which does not give information about interaction between current and backup servers. We need to specify their interaction and study its behavior for a network with multiple servers and under condition of server crash. MARS protocol has been studied and issues have been identified that affects the cluster size - the scalability can be studied by incorporating the lower layer mechanisms to allow the server to decide whether it can add new leaf node to its cluster control VC depending on certain conditions. This requires extension of the current model and also, the current method used in STRESS.

References

- [1] A. Helmy and D. Estrin, "Simulation based 'STRESS' Testing Case Study: A Multicast Routing Protocol", MAS-COTS'98, July 1998.
- [2] A. Helmy, D. Estrin, S. Gupta, "Fault Oriented Test Generation for Multicast Routing Protocol Design", FORTE-PSTV XVIII, 1998 IFIP TC6/WG6.1 Joint International Conference, Paris, France, November 1998.
- [3] A. Helmy, D. Estrin, S. Gupta, "STRESS Testing using Reduced Reachability Analysis: A Case Study for a Multicast Routing Protocol".
- [4] A. Helmy, D. Estrin, S. Gupta, "Systematic Testing of Multicast Routing Protocols: Analysis of Forward and Backward Search Techniques", Proceedings of IEEE ICCN, October, 2000.
- [5] A. Helmy, D. Estrin, S. Gupta, "Systematic Testing of Multicast Routing Protocols: Analysis of Forward and Backward Search Techniques", to be published
- [6] C. Perkins, "IP Mobility Support", RFC 2002, October 1996.
- [7] C. Perkins, "Route Optimization in Mobile IP", Mobile IP Working Group, INTERNET DRAFT, Feb'99. draft-ietf-mobileip-optim-08.txt
- [8] D. Crocker, "ATM Signaling Support for IP over ATM", Network Working Group RFC 1755
- [9] E. Clarke and J. Wing "Formal Methods: State of Art and Future Directions", ACM Workshop on Strategic Directions in Computing Research, Vol. 28, No. 4, pages 626-643, December 1996
- [10] F. Lin, P.Chu, and M. Liu, "Protocol Verification using Reachability Analysis", Computer Communication Review, Vol 17, No. 5, 1987.
- [11] G. Armitage Bellcore, "Support for Multicast over UNI 3.0/3.1 based ATM Networks", Network Working Group RFC 2022, November 1996.
- [12] G. Armitage, "Issues affecting MARS Cluster Size", Network Working Group RFC 2121,, March 1997
- [13] M. Abramovici, M.A Breuer, A.D. Friedman, "Digital Systems Testing and Testable Design", 1990, Computer Science Press.
- [14] R. Cole, D. Shur, C. Villamizar, "IP over ATM: A Framework Document", Network Working Group RFC 1932,
- [15] Zhe Dang and Richard A. Kemmerer "Using the ASTRAL model checker to analyze mobile IP"; Proceedings of the 1999 international conference on Software engineering , 1999, Pages 132 - 142