Achieving Resilience of Heterogeneous Networks Through Predictive, Formal Analysis

Zhijing Qin* Grit Denker† Carolyn Talcott† Nalini Venkatasubramanian*

*(zhijingq,nalini)@ics.uci.edu †(grit.denker,clt)@sri.com

*Department of Computer Science
University of California, Irvine
Irvine, CA, USA

†Computer Science Laboratory
SRI International
Menlo Park, CA, USA

ABSTRACT

Rapid development and wide deployment of wireless technologies in recent years have brought an increasing number and variety of services that are accessible directly from mobile terminals via multiple network access technologies (e.g, Ethernet, WiFi, Bluetooth, LTE, etc). A particular traffic flow may go through different kinds of networks, which greatly increases the end-to-end connectivity opportunities. However, the disadvantage of multinetworks is that a failure or change in one network type may affect many traffic flows. Thus, the various networks in a multinetwork cannot be managed in isolation. Rather we need methodologies that analyze the effects of changes in these dynamic and heterogeneous network environments in unison. Traditional network analysis approaches only focus on static network attributes and do not fully consider the impact of failures on quality of services (QoS) across flows. In this paper, we design and implement a "what-if" analysis methodology using formal methods. Our methodology analyzes the impact of failures and changes in heterogeneous networks on QoS of flows. The results of the formal analysis can guide network administrators in their decisions to proactively adapt network configurations to achieve mission or application objectives. We illustrate our methodology with the help of use cases such as incorporating additional nodes in a network or reconfiguring the network due to failure. We compare our results with conventional network configuration approaches and show how our formal methodology provides more effective decision support than conventional network configuration approaches and that it scales better than simulation approaches.

Categories and Subject Descriptors

I.6.4 [**Model Validation and Analysis**]:

Keywords

Formal Analysis, Heterogeneous Networks

HiCoNS'13, April 9–11, 2013, Philadelphia, Pennsylvania, USA.

1. INFRASTRUCTURE DEPENDABILITY IN MULTINETWORKS

Instrumented Cyber Physical Spaces (ICPSs) are physical spaces that have been instrumented with "intelligence" through heterogeneous sensing, actuation and communication mechanisms. Sensing capabilities range from heat, pressure, movement, temperature, to images or videos to name a few. Devices can be actuated to achieve desired changes in the physical world. ICPSs are relevant for many application domains, including infrastructure security, health care, or smart homes. The power of ICPSs lies in achieving a common picture of the physical space–also called situational awareness (SA)–by observing sensors in unison and deciding how to best act upon that SA to achieve application, human or mission needs. We illustrate our vision of adaptive ICPSs with a campus emergency response scenario. Envision a university campus instrumented with cameras overlooking various areas and monitoring building hallways. Temperature sensors are installed throughout buildings to monitor for situations such as fires. Movement sensors observe pedestrian traffic and can determine whether areas are vacated or not. Mobile devices provide location information for individuals and can be a source of videos or images.

The ICPS can be viewed as a *multinetwork* consisting of devices that have a variety of communication capabilities and mobility profiles. Communication services range from Ethernet to sensor networks to wireless services such as Bluetooth, Zigbee, 802.11 or WAN. Achieving SA in such multinetworks to adapt them to mission needs is challenging. Take for example the case of a fire. Some of the cameras or the networks they are using to communicate data may no longer be operational. It is necessary to re-orient other cameras to ensure surveillance of the entire space. If the network has been partially rendered unusable through a fire, traffic may need to be rerouted or additional mobile routers might need to be deployed to support the network load. Increased traffic load may be due to first responders needing to coordinate their actions. All these changes will cause the network load to change and network administrators require information for decision making - networks may need to be reconfigured and additional network components may need to be deployed in order to best suit application needs under changed circumstances. This is but one example of an *ICPS multinetwork that can be dynamically adapted to new infrastructure situations or application quality-of-service (QoS) requirements.*

Our overall objective is to provide techniques for dependability and resilience in multinetwork ICPSs. While there are potential benefits in performance and QoS of Multinetworks, there are also potential new challenges. if one network goes down, it might affect various network traffic flows. Therefore, it is not longer sufficient to manage each network in isolation. Rather network administrators have to manage networks in the multinetwork in unison to achieve best performance across all network traffic and networks.

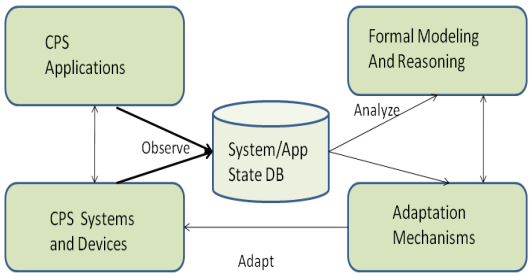


Figure 1: A Reflective CPS Architecture

We propose a reflective (observe-analyze-adapt (OAA)) architecture as shown in Figure 1 to achieve this objective. At the heart of the OAA paradigm is a database with system and application state information. This digital state information of the ICPS guides a range of "safe" adaptations to achieve end-to-end infrastructure and information dependability. There are many challenges that need to be solved to achieve this vision. In this paper we address one of those challenges: provide **resilience and dependability of ICPSs in the face of network or infrastructure failures**. We propose an approach to proactively deal with disruptions to the communication infrastructure of multinetworks. We use the OAA approach because the multinetwork has a model of itself, its objectives, and its effects on the environment. Our goal is to achieve network dependability and resilience through adaptation using runtime application of formal analysis methods.

Our work is set in the context of a multinetwork management platform called "Multinetwork INformation Architecture" (MINA). MINA (see Figure 2) is a middleware system developed on top of UC Irvine I-Sensorium [13]. MINA collects network state information from heterogeneous networks and provides management functions to address resource provisioning, configuration management and fault analysis.

MINA is organized into a three-tier architecture with mobile, but resource-constrained nodes in Tier 3, more stationary but also more powerful nodes in Tier 2, and a centralized server in Tier 1. Different kinds of networks are present in a multinetwork ranging from wired networks (solid lines in Figure2) to wireless networks (dashed lines in Figure 2). Accordingly nodes differ in their capabilities (mobility, computation, memory, power, etc) and edges in their link characteristics. Nodes may be equipped with more than one access technology.

The objective of our analysis is to ensure that quality-of-service (QoS) requirements of network flows are satisfied. A network flow is traffic flowing from a source node to one or more target nodes. For example, a flow may describe imagery flowing from a camera installed on campus to the campus' security control center. Or a flow may describe the traffic that flows from the control center to all cameras in a building to disable power-safe modes on the cameras. The concept of network flow abstracts from the specific route the packets for this flow will take. To check satisfaction of flow QoS, we need network state information as collected in MINA. We employ formal analysis methods to check QoS satisfaction.

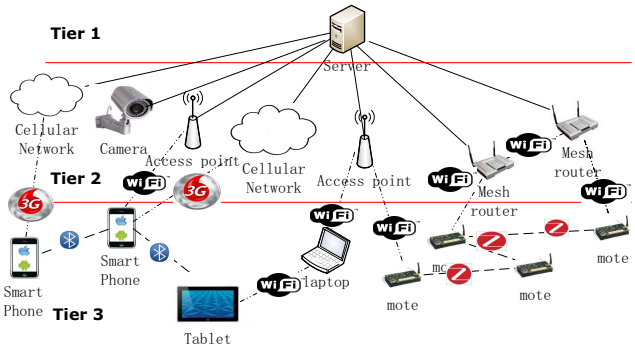


Figure 2: MINA: A Multinetwork Information Architecture

The proposed analysis methodology performs various "what if" analyses asking questions such as

- What happens if a node fails? The objective of this analysis is to determine how critical a node is, not just to one network flow, but to all flows.
- What should we do if we have additional network resource such as mobile routers? Where should we deploy them most effectively? The objective of the analysis is to determine where additional nodes would positively impact QoS of all flows.
- What happens if the load of selected flows changes and how can we best mitigate the effects?
- What happens if the link quality changes due to congestion or interference and how can we best mitigate the effects of degraded link quality?

Answers to these questions help to decide how to best utilize network resources or reconfigure the network to achieve better overall QoS requirements satisfaction. In turn, this will lead to better network resilience and dependability.

Related Work. Existing work on network analysis and prediction can be divided into static analysis and dynamic analysis. Static analysis only exams a snapshot of the the network. In [16], the authors determine the full network state including Internetwork Operating System bugs of devices, configuration errors, static/dynamic routing, and so on. They calculate all possible virtual paths according to the full network state and then compare the virtual paths with the available physical paths to check reachability. However, due to the large search state space, this approach is not very efficient. The same problem exists in [10]. The authors precompute routing tables for each state and employ formal methods to model every possible behavior of the network. [6] provides an effective way to reduce the state space. Both [10] and [16] only focus on the one time link failure and the network will converge to another stable state. In dynamic analysis a failure may cause other failures in future

states. For example, a failing node does not only affect the flows going through that node, but also other flows due to flow rerouting. This is called cascading failures. In [8] the authors consider statistics of link failures and limited number of cascading failures. Different failures combinations can lead to loss of connectivity within a network or to severe congestion, as shown in [12]. They proposed a framework to analyze link availability in the context of link failures, changes of user behavior and routing. However, most of the cited approaches do not consider heterogeneous networks and do not study how the flows are affected by failures. In this paper, we provide an approach to handle (1) changing heterogeneous network topology (i.e. node failure, adding backup router, or network reconfiguration), while (2) taking into consideration how the flows' QoS performance are affected due to flow redistribution triggered by changes in the network topology. This approach is based on formal methods. Formal methods have been used to do network analysis [1], protocol proof [15] and network model checking [11]. To our best knowledge, there exist no tool that uses formal methods for the flow oriented predictive analysis of multinetworks.

The remainder of this paper is organized as follows: We introduce our formal model in Section 2. In Section 3 we present three use cases and compare the results our analysis method with conventional methods. Section 4 closes with concluding remarks.

2. FORMAL METHOD-BASED ANALYSIS METHODOLOGY

The objective of our analysis is to answer questions such as "what effect does a failing node have on all network flows" or "if one had more resources to deploy in the network, what would be the best location for an additional node." It turns out that underlying all of the questions listed in Section 1 is the concept of how critical a node is to the satisfaction of QoS requirements. We introduce the notion of "Node Criticality Index (NCI)." Nodes with the highest criticality index have the most negative effect on the overall QoS of flows when they fail. Thus, these nodes are the first targets when it comes to deploying backup nodes or relieving a node of high traffic loads. We distinguish our concept of NCI from conventional node importance measures used in existing work. The simplest node importance measure is node degree, which is defined as the number of neighbors a node has (see [2]). Closeness-based measure is another node importance measure that finds the distance center or the median of a graph. It has application in facility location [9], package delivery [3] and operations research problems. It is computed by summing up the distances from the current node to all remaining nodes. Betweenness is one of the most prominent node importance measures. It measures the influence of a node over the connection of other nodes by summing up the fraction of shortest paths between the other nodes that pass through it [4, 7]. However, none of these measures take into account how redistributed flows due to network changes influence overall QoS. The analysis methodology proposed here addresses that issue.

We use Maude [5] as our underlying formal method tool. Maude is a multiparadigm executable specification language encompassing both equational logic and rewriting logic. Maude allows modeling system states through user-defined data types and system dynamics through rewrite rules. The Maude interpreter is very efficient, allowing prototyping of quite complex test cases. Maude also provides efficient built-in search and model checking capabilities. Maude sources, executables for several platforms, the manual, a primer, cases studies, and papers are available from the Maude website `http://maude.cs.uiuc.edu`.

We formalize the analysis objectives as Maude models and rules. For example, if the analysis objective is to determine the most critical network node, then we provide a Maude specification and a set of rules that simulate in a network that a node fails, reroute affected flows and simulate QoS of the new network. Iterating this process over all nodes and comparing the resulting QoS of all flows allows us to determine the most critical node. If the analysis objective is to choose the best new access point for a wireless node when its access point fails, then we have a Maude specification and rules that simulates the overall performance for the different alternate access points. Thus, we generate a Maude specification according to the reasoning objective (step 3 in Figure 3). Information about the specific network topology and state and the flows is pulled from the MINA database (step 2 in Figure 3). We use the Maude engine to execute the specific analysis (step 4) and store any relevant analysis results (e.g., node criticality index or hints where new resources should be deployed) in the database (step 5). This information can be used by network administrators to devise new network reconfigurations and issue accordingly commands to the multinetwork (step 6).

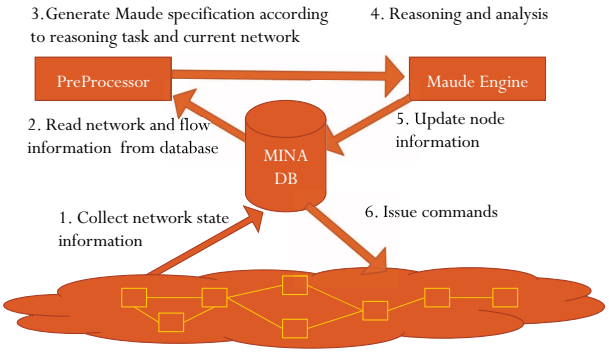


Figure 3: Workflow in the context of MINA

In the following we present some details about how we model networks and flows and what the rewriting rules to process network flows and determine QoS look like.

2.1 Network and Flow Specification

Information about networks and flows is stored in the MINA database as follows. A network node is represented by a 3-tuple containing *NodeId*, *neighbor list* and *weight list*. *neighbor list* contains all nodes directly linked to the node *NodeID*. The ith element in *weight list* represents the capacity of the link between *NodeId* and the ith element in the *neighbor list*. A flow is represented by an 8-tuple: *source*, *route*, *destination*, *flowID*, *type*, *throughput*, *packetlength* and λ. *route* is the sequence of nodes on the path of the flow from the source to the destination, *type* specifies the kind of traffic (e.g., ftp, audio, video, and so on), *packetlength* corresponds to the packet length when it is an ethernet packet or the length of the item sent on a different network, and λ is the mean

packet arrival rate of this flow. Note the product of λ and *packetlength* is the throughput. In the following, we will show how we use the information about networks and flows that is stored in the MINA database to generate a Maude model.

We consider the nodes and links through which a flow goes as tandem queues (see Figure 4). The tandem queues refer to an arrangement of queues in which the queues are lined up one after the other: the outgoing traffic of the current queue is the incoming traffic of the next queue. A node consists of the queue and the network interface (NIC; there is possibly more than one NIC on a node). Currently we do not consider queue length limitations, or in other words, we assume there no packets are lost due to limited queues.

Since link propagation delay is extremely small, we only consider delay introduced by queueing and service time in the NIC. Thus, in our abstract model we assume the outgoing packet of one node to be the incoming packet of the subsequent node. This way, we divide an end-to-end flow into several sub flows, one for each hop along the path of the flow (see Fig 4). By doing so, we can iteratively propagate the QoS parameter computed in each node and accumulate the end-to-end QoS for a flow from its source to its destination. In this paper, we use delay as an example QoS parameter to illustrate our analysis methodology.

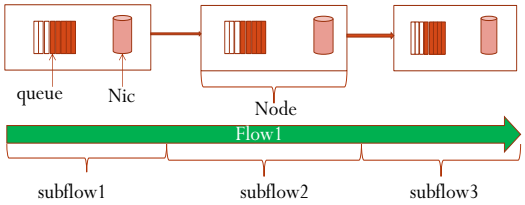
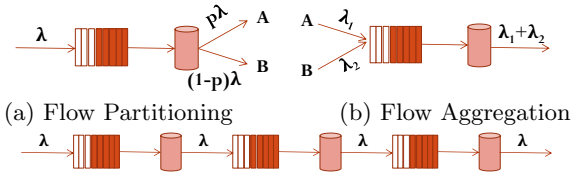


Figure 4: Network Flow Model

The flow model enables us to get the end-to-end delay by aggregating and propagating the delay incurred by each node. We use a simple M/M/1 model [14] to calculate the delay incurred by a single node. The M/M/1 represents the queue in a system having a single server, where arrivals are determined by a Poisson process and job service times have an exponential distribution. We denote the Poisson Arrival Distribution with (λ, n), where λ is the packet mean arrival rate and n is the packet length. The server's (network interface) mean service time is denoted by Ts. We use Poisson distribution to describe the flow pattern mainly because it is widely used to simulate network flow [14] and the Poisson distribution has perfect aggregation and partitioning attributes. For example, as shown in Fig 5(c), the Poisson pattern is not changed when a flow goes through a node with exponential service time. In Fig 5(b), flow A and B with mean arrival rate λ_1 and λ_2 are both the incoming Poisson flows at one node. After processing, the new aggregated flow is still a Poisson flow with a mean arrival rate $\lambda_1 + \lambda_2$. In Fig 5(a), a Poisson flow with mean arrival rate λ can be partitioned into two Poisson flows whose mean arrival rate's sum is still λ.

These important features enable us to apply the Possion-based delay calculation in each processor, regardless of how



(a) Flow Partitioning (b) Flow Aggregation

(c) Tandem Queue with Poisson Flow

Figure 5: Key Attributes of Flows with Poisson Arrival

the flows are distributed in the multinetwork:

$$Delay = n/(Ts - n)$$

2.2 Rewriting Rules for Analysis

We view the network as a set of nodes and flows. Each flow is a sequence of subflows, one subflow for each hop along the flow's path. Nodes and (sub)flows are basic concepts in our Maude model. We model the network as nodes N and sub flows SF. We also model the QoS parameters such as $d$elay, throughput, jitter, packet loss and so on. So a network is modeled in Maude as a structure with the following elements: `N(`*`node attributes`*`), SF(`*`subflow attributes`*`),` *`QoS attributes`*.

Nodes N are modeled as a collection of attributes. Each attribute is represented as label(value)-pair. Variables (using capital letters) are used as a place holder for values in structures or rules. Attributes of nodes are the identity of the node (`Nid(N1)`), type of NIC (`NIC(T)`)[1], transmission rate `Tx(X)`), Poison parameters (`PoiPara(`λ`,n)`), the number of flows expected to route through this node (`Tot(S)`), and the flows that have already arrived (`Cur(C)`). Thus a node is represented as `N(Nid(N1),NIC(T),Tx(X),PoiPara(`λ`,n), Tot(S),Cur(C))`.

Each subflow is modeled as a collection of attributes: identity of the parent flow `Par(F1)`, the identity of the subflow `Sub(S1)`, the identify of the source node of this subflow `Src(N1)` and its NIC type (`SrcNIC(T)`), and identity of the destination node of this subflow `Dest(N2)`, the poison parameters `PoiPara(`λ_{in}`,n)` for this subflow. and the incoming and outgoing QoS characteristics of this subflow. Since the examples in this paper only explore delay, we will only model incoming and outgoing delay (`Delayin(DIN)` and `Delayout(DOUT)`). Finally, we add a boolean flag to indicate whether the subflow was already processed by a rule (`Proc(B)`). Thus a subflow is represented as `SF(Par(F1), Sub(SUB1),Src(N1), SrcNIC(T), Dest(N2), PoiPara(`λ_{in}`,`n_{in}`), Proc(B), Delayin(DIN), DelayOut(DOUT))`.

We have three main Maude rules that describe propagation of QoS characteristics of flows in the multinetwork: a) flow accumulation, b) flow processing, and c) flow propagation.

- *flow accumulation:* This rule is collecting all the sublows at one processor. The idea is that when multiple Poisson flows come into a processor, we aggregate the λ and n value for all flows to calculate the overall delay they have suffered given the service capacity of this processor (the capacity/transmission rate of the net-

[1]For simplicity and because of space limitation, we present our rules with one NIC per node.

work interface).

```
crl  N(Nid(N1), NIC(T), PoiPara(λ,n), Cur(C),...)
     SF(Src(N1), SrcNIC(T), PoiPara(λ_in,n_in),
        Proc(false), ...)
  => N(Nid(N1), NIC(T), PoiPara(λ',n'), Cur(C1),...)
     SF(Src(N1), SrcNIC(T), PoiPara(λ_in,n_in),
        Proc(true),...)
 if  C1 := C + 1
     λ' := λ_in + λ
     n' := ( λ_in *n_in + λ * n)/(λ_in + λ).
```

Once the number of the accumulated flows reaches a predefined amount (indicated in processor by the value of attribute *total*), the *flow processing* rule will be triggered.

- *flow processing:* Once all incoming flows are accumulated, the node has a complete set of values of λ and n, and hence is able to calculate the new delay it incurred via the following rule. The new delay is added to the existing delay. Although we only focus on delay, we could similarly calculate other QoS attributes (e.g. jitter, packet loss, etc).

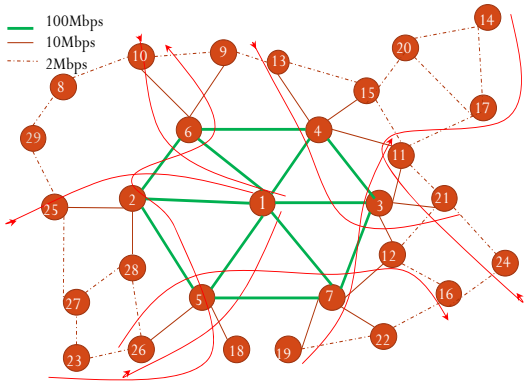
```
crl  N(Nid(N1), Tx(X), PoiPara(λ,n),...)
     SF(PoiPara(λ_in,n_in), Delayin(DIN)
        DelayOut(UNKNOWN),...)
=>   N(Nid(N1), Tx(X), PoiPara(λ,n),...)
     SF(PoiPara(λ_in,n_in), Delayin(DIN)
        DelayOut(DOUT),...)
 if  ( X - λ * n ) > 0
     DOUT := n_in / ( X - λ * n ) + DIN
```

- *flow propagation* Once the outgoing flow is generated by the node via the *flow processing* rule, it will be passed into the next node as an incoming flow. This way, the delay value can be propagated from the source to the destination.

```
rl   SF(Par(F1), Sub(S1), Src(N1), Dest(N2),
        Delayout(DOUT),...)
     SF(Par(F1), Sub(S2), Src(N2), Dest(N3),
        Delayin(UNKNOWN),...)
=>   SF(Par(F1), Sub(S1), Src(N1), Dest(N2),
        Delayout(DOUT), ...)
     SF(Par(F1), Sub(S2), Src(N2), Dest(N3),
        Delayin(DOUT), ...)
```

2.3 Preliminary Validation

We performed a preliminary experiment to validate our Maude model. For this, we used the example network and flows shown in Fig 6(a) and Fig 6(b) as our test scenario. The example network has 30 nodes connected via links with different capacities (i.e. Ethernet 100Mbps, WiFi 10Mbps, BlueTooth 2Mbps), and 8 flows are specified with different arrival rate and packet lengths. Flow has two types: web/ftp (type 1) and audio/video (type 2). We compare end-to-end delays of flows calculated by Maude with those computed by Qualnet simulator where Qualnet models the same network and flows. In Qualnet, we use point-to-point links to simulate the links in Fig 6(a) and we use constant bit rate applications to simulate the flows in Fig 6(b). The results of simulating end-to-end delay (see Fig 8) show that the delays obtained



(a) network

route	flowid	type	TP	Item length (Mbits)	Arrival num (/s)
1,2,25	1	1	0.8	0.0016	500
1,5,26	2	1	0.8	0.002	400
19,7,3,11	3	1	0.8	0.0016	500
21,3,4,13	4	1	0.48	0.0016	300
1,6,10	5	2	0.48	0.0016	300
24,21,11,17,14	6	2	0.4	0.002	200
23,26,5,2,6,10	7	2	0.4	0.002	200
26,5,7,12,16	8	2	0.24	0.0008	300

(b) Flows

Figure 6: Preliminary Experimental Settings

for each flow (x-axis of Fig 8) by Maude are quite consistent with those determined by Qualnet, with an average error of less than 9%. Note that the objective of our formal model is **not** to have the exact same results as Qualnet, but rather achieve enough accuracy so that our prediction results will be viable, while at the same time our analysis method will be more efficient due to its abstraction level. We intend to determine with our formal methods analysis the trends of the delay performance in heterogeneous multinetworks and various kinds of flows. The preliminary experiment has validated our methodology. Next we further show that our methodology can provide solutions for three different case studies.

3. CASE STUDIES

We present three use cases. Each use case differs in its objective to adapt a multinetwork in the face of network failures. All use cases have application QoS requirements that must be satisfied. Different "what if" questions will provide guidance on how to best achieve QoS requirements. To compare the effectiveness of our formal method-based analysis approach with other approaches, we apply conventional strategies in each of the three use cases. We see that our analysis results suggest different adaptations than conventional analysis methods. We compare the suggested adaptation strategies from our analysis and conventional strategies by simulating the resulting networks to assess how well the adaptation strategy satisfies QoS requirements. We see that our analysis methodology provide superior guidance compared with conventional strategies.

3.1 Case 1: Node Criticality Index

Node Criticality Index (NCI) is an indicator of how important a node is. Generally the importance of a node can be measured by the impact of a node's failure on other nodes

or network traffic. In this case study, we measure how end-to-end delays of flows are affected given a node failure. The node whose failure will cause the biggest end-to-end delay is the most important node. Traditional Node Critical Index approaches only focus on static attributes, e.g. node degree [2], centrality [9, 3], or workload/betweeness [4, 7]. These traditional measures are not taking into account the affect that a node failure will have on the flows that go through the failing node, and thus, the impact a failing node has on other flows due to contention caused by the redistribution of flows.

Our "what-if" analysis methodology and tool assumes that each node fails, one at a time, and generates a new network and new paths for the flows (using the Dijkstra algorithm to redistribute the flow, which is widely employed in routing protocols). We restrict failing of nodes to those nodes that are not source or destination of a flow. The reason is that if we would let those nodes fail, then we would not only change the paths of flows, we would also change the set of flows. And this would not allow us to compare the end-to-end delay results we generate for one node failing at a time. There are also nodes that are on the critical path of flow, meaning if these nodes fail, there is no longer a path between source and destination of at least one flow. Again, we assume these nodes do not fail so that we can consider the impact of a node failing under the same traffic load or set of flows. Thus, in summary, our analysis lets one node at a time fail—for those nodes that are not source, destination or critical path nodes—and compares the impact of a node's failure on all flows to compute which one has the most negative influence when failing. That node is deemed to be the most critical node.

In the following we refer to nodes that we do not let fail as *critical nodes* and those that we let fail as part of the analysis as *measurable nodes*. We are using the network and flows from Figure 6. The left table in Figure 7 below summarizes the average end-to-end delays on all eight flows in terms of the delay increase/decrease multiplication factor. The criticality rating is derived from the average multiplication factor, rating the most critical node to be the one with the biggest positive multiplication factor. We only show the top six most important nodes as determined by our Maude analysis (left table) or as determined using traditional approaches based on degree and workload (right table). For

Failed Node	Ave	Rank
Node 2	4.2	2
Node 3	4.4	1
Node 4	1.4	5
Node 5	1.1	6
Node 6	1.6	4
Node 7	1.9	3

Importance Ranking by Degree and load			
node	degree	load	Rank
5	6	1.44	1
3	6	1.28	2
7	6	1.04	3
4	6	0.48	4
2	5	1.2	5
6	5	0.88	6

Figure 7: Node Importance Ranking: Maude Based Results (left) vs. Degree/Workload Based Results (right)

example, when node 2 fails, the average end-to-end delay on all flows will be increased by 4.2 times, and when node 3 fails, then the multiplication factor is 4.4. In comparison, we get a different ranking with the traditional approach as shown in the right table of Figure 7.

To determine which analysis method delivers a better quality answer to the question "which node is the most important node?", we devised another experiment. The two most important nodes according to the "what-if" Maude analysis are node 3 and node 2 (labelled Maude Group in Fig 9), while traditional degree and workload-based analysis (labelled MaxDegree Group in Fig. 9) determine node 5 and node 3 to be the most critical ones. The next experiment assumes that the network administrator followed either the Maude analysis results of the MaxDegree analysis results and backed up those two most important nodes. Thus, it is very unlikely for those nodes to fail, but the remaining 16 nodes can still fail. For each group we model the networks in Qualnet and let the 16 nodes each fail for 300s. This means, there is a 4800s long experiment in Qualnet of each group (Maude Group and MaxDegree Group). The two 4800s experiments have the same original flow configuration. In addition, we run another experiment of 4800s without any node failures. We compare the end-to-end delay of the Maude Group and MaxDegree Group with the original network and flows (without any failure), to see how much increase in flow delay we get for either group. Fig 9 shows that if we protect the two modes selected by our "what-if" analysis tool, the average delay increase percentage is smaller (i.e., 25%) than if we protected the nodes chosen by a MaxDegree-based analysis (i.e., 34%)(Although some flows in MaxDegree group do not incur extra delay, i.e. 2, 6, 8). Thus, our "what-if" analysis is able to provide a more accurate NCI for this scenario.

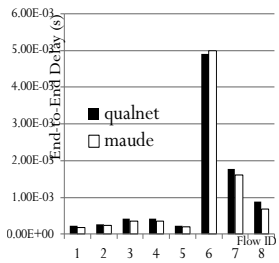


Figure 8: Preliminary Experimental Results

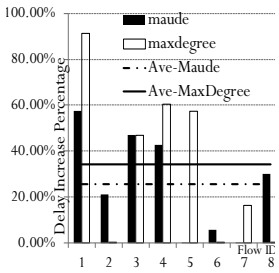


Figure 9: Delay increase (in %) for the two analysis approaches

3.2 Case 2: Adding an Additional Router

If one has limited resources available, it becomes a challenge to place them in the network in a way that ensures best use of the resource's capabilities. This is a common problem for network planning and administration. For example, if we have only one extra node which happens to be a router, where should we put it so that the overall end-to-end delay for all flows will be improved most? Traditional approaches usually put the extra router next to the node with the biggest workload. However, adding a new router will cause flow redistribution which will in turn impact the end-to-end delay of all flows. This aspect is usually not taken into consideration in traditional approaches. In our MINA architecture, wireless routers are in Tier 2 (Tier 1 is the centralized server and Tier 3 contains the mobile nodes), hence we determine that there are six tier two nodes (nodes 2-7)

in Fig 6(a) that can be further supported by an additional router. Placing an additional router next to a node we assume that this router will have the exact same neighbors and links as that node. However, adding such extra router into the network means that the network changes and thus, flow routes will change too. Our "what-if" analysis tool examines end-to-end delay of each flow for possible position of the additional router and compares the results to determine the optimal placing of the additional router. We use again the network in Fig 6(a) and the flows in Fig 6(b) with exception of the differences indicated in Fig 12.

Flow ID	Ave Delay Inc.	Rank
2	98.71%	1
3	99.25%	2
5	99.61%	4
6	99.50%	3
7	99.99%	5

Figure 10: Best Position for additional router: Ranking by Maude analysis.

Node	Contention link	Flow ID	Load	Load sum	Rank
2	2-25	1	0.8	1.2	2
	2-28	9	0.4		
3	3-11	3	0.8	1.28	1
	3-21	21	0.48		
5	5-26	2	0.8	1.04	3
	5-26	8	0.24		
6	6-10	5	0.48	0.88	5
	6-10	7	0.4		
7	7-3	3	0.8	1.04	3
	7-5	8	0.24		

Figure 11: Best Position for additional router: Ranking by Workload-based analysis.

In Fig 10 we can see that when adding the extra router next to node 2, then the average end-to-end delay on all flows is 98.71% of the one without extra router. Thus this position has the best end-to-end delay improvement, according to our "what-if" analysis. In comparison, if traditional workload-based approach is used, the best position is node 3, as shown in Fig 11. Note: Node 4 is not in the Fig 11, because in order to do load balancing, nodes need to have at least two flows, and Node 4 only has one.

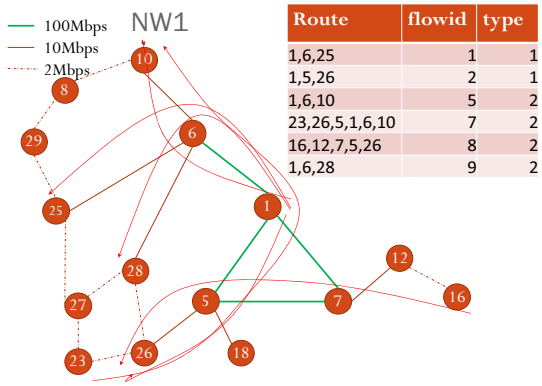
route	flowid	type	TP	Itemlength	Arrival num
13,4,3,21	4	1	0.48	0.0016	300
16,12,7,5,26	8	2	0.24	0.0008	300
1,2,28	9	2	0.4	0.0008	500

Figure 12: Flows in Case 2

Next we deploy the extra router next to nodes 2 and 3, respectively, and use Qualnet to test end-to-end delay improvement. The results show that the average delay decrease rate over all flows is 1.43% if we position the extra router next to node 2, while it is 0.01% if we position it next to node 3. Hence we can conclude that if we only have one extra router, node 2 is the better position to place, which is consistent with Maude "what-if" analysis.

3.3 Case 3: Network Reconfiguration

Mobile devices usually have more than one WiFi hotspot with which to associate. Properly selecting an access point is good for network resource utilization and application performance. We use again the network and flows as in Fig 6. If node 2 is down, node 25 and node 28 need to re-associate with either node 6 or node 5. Hence we have four network reconfiguration plans as determined by the possible combinations for node 25 and 28 re-associating with node 5 or 6. One of the four network reconfigurations and the new routes for flows are shown in Fig 13. We refer to NW1 where both node 25 and node 28 re-associate with node 6 (25-6;28-6). NW2 is the network where node 25 re-associates with node 6 and node 28 re-associates with node 5 (25-6;28-5). NW3 is (25-5;28-5) and NW4 is (25-5;28-6).



Route	flowid	type
1,6,25	1	1
1,5,26	2	1
1,6,10	5	2
23,26,5,1,6,10	7	2
16,12,7,5,26	8	2
1,6,28	9	2

Figure 13: One Possible Network Reconfiguration when Node 2 goes down

The "what-if" analysis tool calculated the end-to-end delay of flows for all four reconfigurations (see Fig 14). The results of the Maude analysis show that NW2 and NW4 will have less delay degradation compared with the one in original network (node 2 is on). However, if we use traditional load balance-based approach (called *LoadBalance*), NW3 and NW4 are the better choice, as shown in fig 15.

Network Config	Avg Delay Increase	Rank
NW1	54.0502%	3
NW2	27.6939%	1
NW3	65.6767%	4
NW4	28.5294%	2

Figure 14: Best Reconfiguration Ranking using Maude

Network config.	Node6 load:	Node5 load:	Delta	Rank
NW1	2.26	1.04	1.22	4
NW2	2.16	1.14	1.02	3
NW3	1.36	1.94	0.58	2
NW4	1.46	1.84	0.38	1

Figure 15: Best Reconfiguration Ranking by Workload

We then put these four resulting configurations into Qualnet to get the average end-to-end delay over all flows. As Fig 16 shows, Maude results are quite consistent with Qualnet results. I.e., Qualnet would prefer NW2 and NW4 over NW1 and NW 3, so does Maude. LoadBalance would prefer NW 4 and NW 3 over NW 1 and NW 2. Note the end-to-end delay generated by Qualnet is not exactly the same as the one generated by Maude "what-if" analysis tool. However, the trends and the relationship among different network configurations are quite similar.

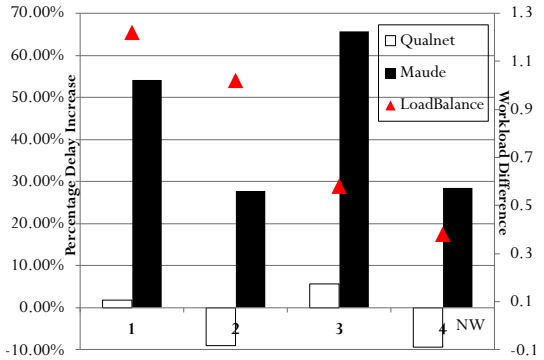


Figure 16: Verification by Qualnet

Note we use Qualnet as benchmark because it is a well known network simulator that can accurately describe the network behaviors across multiple layers. However, we cannot use Qualnet to replace "what-if" analysis tools. Maude is a high-level language providing conceptual abstractions. The three rules in our tools are the basic elements derived from the flow and queue models. Once these basic elements are in place, it is easy to set up and carry out analysis on various network without any further configuration. Network simulators usually need to simulate the packet/frame behaviors in all layers, which is more time-consuming. Moreover, having a human setting up these networks in Qualnet requires time and introduces the potential of misconfiguration by the human. In addition, increasing packet number means more calculation and time while our formal-method based tools scales up nicely due to abstraction. In Fig 17, we take the "No additional routers" scenario from case 2 and the "No failure happened" scenario from case 3 as examples to illustrate the efficiency and scalability of our Maude based "what-if" analysis tool. Maude only use 27ms and 11ms in two scenarios, respectively. Qualnet uses much more time in our experiments. Even if we only look at the time consumption in one second-this is the time used to simulate the same data amount with Maude, Qualnet still consumes more time. When we increased the number of packets by a factor of 10, then the time consumed by Maude is constant while it increases in Qualnet. Hence we argue that our formal language based "what-if" analysis tools have better scalability.

	Maude	Qualnet	Qualnet with 10 times packets
Case2: bp0	rewrites: 481 in (27ms real)	20.4160s/300s =68ms/1s	24.2847s/300s
Case3: nw0	rewrites: 295 in (11ms real)	12.9815s/300s =43ms/1s	14.8934s/300s

Figure 17: Time consumed by Maude and Qualnet

4. CONCLUSION AND FUTURE WORK

Traditional network analysis approaches mainly focus on faults and security problems in wired, homogeneous networks where nodes have enough capabilities and links are stable. In addition, those techniques do not take into consideration how changes to the network trigger flow redistribution that potentially impacts QoS of all flows.

In this paper, we proposed and evaluated a novel what-if analysis tool based on the formal language Maude. We modeled heterogeneous networks in Maude using a general flow and queue model and provided rules that compute how end-to-end delay in the network. Using these models, we can evaluate various kinds of network changes (i.e. failures, backup and reconfigurations) and determine improved network configurations. For example, we can determine what is the best position for additional resources or which nodes are most critical when it comes to failures. Experiments from three case studies have shown that our tool outperforms traditional approaches. The what-if analysis tool could be extended to adopt more QoS example parameters and heterogeneous applications in the future. Another area of future work is to investigate flow models for other types of traffic and formalize those in Maude. This would allow us to handle network traffic more comprehensively.

5. REFERENCES

[1] Model-based environment for validation of system reliability, availability, security, and performance https://www.mobius.illinois.edu/.

[2] P. Bonacich. Some unique properties of eigenvector centrality. *Social Networks*, 29(4):555 – 564, 2007.

[3] S. P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55 – 71, 2005.

[4] U. Brandes. On variants of shortest-path betweenness centrality and their generic computation. *SOCIAL NETWORKS*, 30(2), 2008.

[5] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martí-Oliet, J. Meseguer, and C. L. Talcott, editors. *All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

[6] G. de Silva, P. Matousek, O. Rysavy, and M. Sveda. Formal analysis approach on networks with dynamic behaviours. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress on*, pages 545 –551, oct. 2010.

[7] E. Estrada, D. J. Higham, and N. Hatano. Communicability betweenness in complex networks. *Physica A: Statistical Mechanics and its Applications*, 388(5):764 – 774, 2009.

[8] Q. Gan, Bjarne, , and E. Helvik. Dependability modelling and analysis of networks as taking routing and traffic into account. In *Next Generation Internet Design and Engineering, 2006. NGI '06. 2006 2nd Conference on*, 2006.

[9] P. Hage and F. Harary. Eccentricity and centrality in networks. *Social Networks*, 17(1):57 – 63, 1995.

[10] P. Matouek, J. R, O. Ryavy, and M. Svéda. A formal model for network-wide security analysis. In *Proceedings of the 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, ECBS '08, pages 171–181, Washington, DC, USA, 2008. IEEE Computer Society.

[11] A. Mclver and A. Fehnker. Formal techniques for the analysis of wireless networks. In *Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on*, pages 263–270. IEEE, 2006.

[12] M. Menth, M. Duelli, R. Martin, and J. Milbrandt. Resilience analysis of packet-switched communication networks. *Networking, IEEE/ACM Transactions on*, 17(6):1950 –1963, dec. 2009.

[13] Z. Qin and L. Iannario. Towards design of an overlay architecture in the multinetwork management system. Technical report, University of California, Irvine, http://www.ics.uci.edu/ dsm/cypress/publications.html, March 2012.

[14] W. Stallings. Queuing analysis, 2000.

[15] A. Ten Teije, Marcos, et al. Improving medical protocols by formal methods. *Artificial intelligence in medicine*, 36(3):193–209, 2006.

[16] G. G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford. On static reachability analysis of ip networks. In *in Proc. IEEE INFOCOM*, 2005.