# Fault localization in Communication Networks

Vijay Akkineni
Department of Computer Science and Engineering
Georgia State University
vakkineni1@student.gsu.edu

## Abstract

This paper is about fault localization techniques in communication networks worked for Phd qualifiers examination. Fault Localization is an important aspect of network fault management and is a process of deducing the source of a failure from the set of observed indicatations. It has been an important research area in the field of networking both communication networks and wireless sensor networks. As commuication networks grow in size and complexity it has been imposing new set of requirements on fault localization. Despite the amount of research done in this field we can still argue that it is an area of open for research. The paper essentially disucsses the work that has been done in this field in the pas few years with emphasis on the both the papers given for the examination.

## Categories and Subject Descriptors

H.4 [**Communication Networks, Sensor Network Applications**]: Miscellaneous

## Keywords

Fault Localization, Communication Networks, Root Cause Analysis, Causal Inference

## 1 Introduction

Fault diagnosis is an important part of networking. Faults are unavoidable in communication systems but their quick detection and isolation and repair is critical for the robustness, reliability and health of the system. When the networks get large and cumbersome automatic fault diagnosis and fault management is a crucial aspect.

A basic taxonomy in this field is mentioned below.

Event, defined as an exceptional condition occurring in the operation of hardware or software of a managed network, is a central concept pertaining to fault diagnosis.

Faults (also referred to as problems or root causes) constitute a class of network events that can cause other events but are not themselves caused by other events. Faults may be classified as: (1) permanent, (2) intermittent, and (3) transient. A permanent fault exists in a network until a repair action is taken. Intermittent faults occur on a discontinuous and periodic basis causing a degradation of service for short periods of time. However, frequently re-occurring intermittent faults significantly jeopardize service performance. Transient faults cause a temporary and minor degradation of service.

Error(Failure) is defined as a discrepancy between a computed, observed, or measured value or condition and a true, specified, or theoretically correct value or condition. Error is a consequence of a fault. Faults may or may not cause one or more errors. Errors may cause deviation of a delivered service from the specified service, which is visible to the outside world. Errors do not need to be directly corrected, and in many cases they are not visible externally. However, an error in a network device or software may cause a malfunctioning of dependent network devices or software. Thus, errors may propagate within the network causing failures of faultless hardware or software.

Symptoms are external manifestations of failures. They are observed as alarmsnotifications of a potential failure. These notifications may originate from management agents via management protocol messages (SNMP trap and CMIP EVENT-REPORT), from management systems that monitor the network status, e.g., using command ping, system logfiles or character streams sent by external equipments.

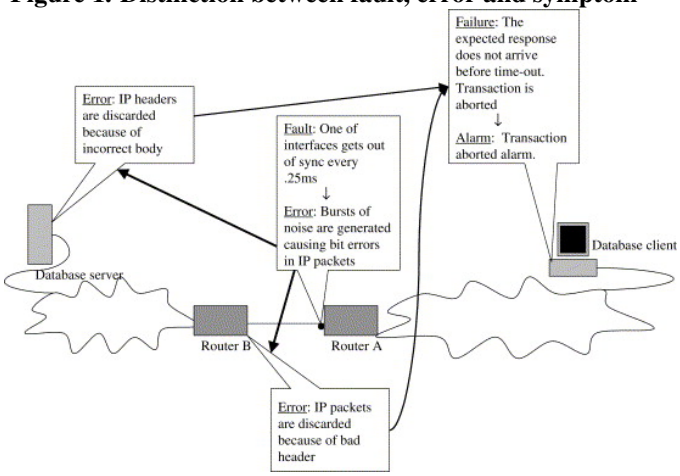
Figure 1 shows the concepts described above in an end to end perspective.

The process of fault diagnosis usually involves three steps as cited in [1]:

- Fault detection, a process of capturing on-line indications of network disorder provided by malfunctioning devices or fault detection agents in the form of alarms.
- Fault localization (also referred to as fault isolation, alarm/event correlation, and root cause analysis) is a set of observed fault indications is analyzed to find an explanation of the alarms.
- Testing is a process that, given a number of possible explanations, determines the actual faults.
- Fault Correction, by which we mean not only to diagnose, but also to repair all faulty components within a

network (This includes the testing component).

**Figure 1. Distinction between fault, error and symptom**




The difficulty in the fault localization process arise from the ambiguity of the observed set of alarms as same alarm can be generated from multiple different faults and multiple alarms can correlate back to the same fault. The incompleteness of the alarm stems from the fact that the alarm doesn't have all the information or there is a loss of alarm. Inconsistency among the observed alarms results from device perception of the fault is different form device to device.

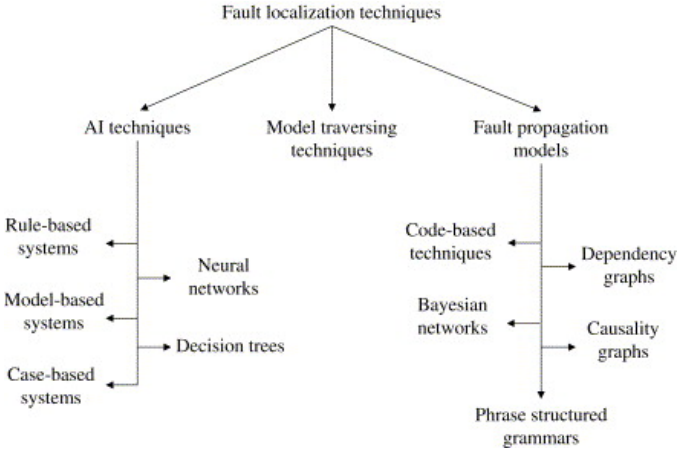
A set of alarms generated by a fault may depend on many factors such as dependencies among network devices, current configurations, services in use since fault occurrence, presence of other faults, values of other network parameters, etc. Due to this non-determinism the system knowledge may be subject to inaccuracy and inconsistency. Fault evidence may also be inaccurate because of spurious alarms, which are generated by transient problems or as a result of overly sensitive fault detection mechanisms. When spurious symptoms may occur, the management system may not be sure which observed alarms should be taken into account in the fault localization process. Event management systems should identify and eliminate multiple simultaneous related or unrelated root causes.

In large networks distributed fault localization process should be performed in distributed fashion. Many researches [2] and [3] have concluded that distributed fault localization using a set of event management nodes is a better approach than centralized process. The complexity arised from the nature of error propogation, they propogate either horizontally to the peers or vertically from bottom layer to upper layers or afecting higher level services. The errors also propogate from one management domain to a different management domain making the process even difficult. Therefore the distribution of complexity to distributed fault localization processes and inferring form the collective process makes the process computationally feasible and efficient.

An alarm can insinuate different type of faults that occurred in different communication devices where in fault localization process may not comeup with a definitive answer. Few approaches that will be discussed in this paper combine fault localization with testing and fault ocrrection process to validate the hypothesis. Therefore there should be some kind of optimality measure or confidence measure that should be employed in measuring the hypothesis that the localization process came up with and it could include the lowest cost or min failure probability or some heuristic function which optimizes the process of validating the hypothesis.

Numerous works have been proposed on fault localization process. The techniques are dervice from different areas of artificial intelligence, graph theory, neural networks, automata theory and many other approaches. Figure 2 broadly classifies the exisiting solutions. The two papers provided for the exam falls into the category of end to end testing scenarios and proababilistic reasoning and derive their roots from bayesian network analysis.

**Figure 2. Classification of fault localization techniques**




## 2 Expert Systems techniques

Most widely used technique in the field of fault localization and diagnosis are expert systems as they try to mimic the actions of human expert. Most expert systems use rule based system as their inference engine. [4],[5] and [6] are examples of expert systems.

The expert systems developed differ in the knowledge they use. Rule based fault localization solely depend on the structure of the knowledge base as rule definition language. In [7] the knowledge base is divided to reusable knowledge modeled as core knowledge and customized knowledge. in [5] the rules are organized as composite events and an intervention of human expert is required to update the event base. The rule based systems can act as a powerful tool to eliminate least likely hypothesis. Although the RBR paradigm is appropriate for problem-solving tasks that are confined and well-understood its limitations are.

- an in-ability to learn from experience.
- Fan inability to deal with novel problems.
- the difxulty of updating the systems to keep up with rapidiy changing domains such as expanding heterogeneous network.

in Model bases expert systems like [6] conditions are ususally accosiated with rules which includes predicates re-

ferring to system model. These predicates test the syetem for existence of relationship among system component. [6] uses correlation tree skeletons describing cause and effect relation ships between event. regardless of what expert system is being used localization process is always driven by inference engine and correaltion rules between events.

[8] and [9] are examples of Case-based reasoning systems. The goals of CBR systems are (i) to learn from experience, (ii) to offer solutions to novel problems based on past experience, and (iii) to avoid expensive maintenance. The basic idea of CBR is to recall, adapt, and execute episodes of former problem-solving in an attempt to deal with a current problem. Former episodes of problem-solving are represented as cases in a case library. When confronted with a new problem, a CBR system retrieves a similar case and tries to adapt the case in an attempt to solve the outstanding problem. The disadvantages of such a system is the time complexity in retrieving a case that is the closest match to the event and the close tailoring of the application to the domain.

In addition to the above mentioned techniques there are other notable techniques are neural networks based approaches [10] [11] and decision tree based approaches [12]. Neural networks have parallel computing architecture and are very fast avoiding bottlenecks which commonly arises from serial processing. But the main disadvantage is that the learning process requires intesive training and in communication networks where all the alarms signatures are not readily available for pattern recognition.

Decision tree aproaches are simple and allows expressive representation of expert knowledge howeever they are limited by the dependencies sepcific applications nad degraded accuracy in noisy scenarios [13] [14].

## 3 Model traversing techniques

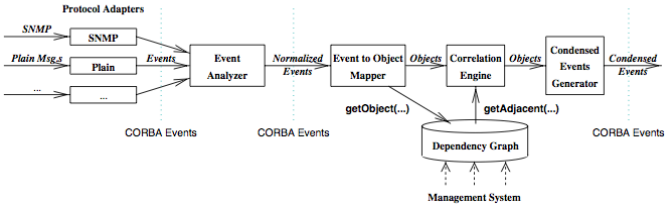
[16], [17], [18] and [19] are all Model Traversing that use formal representation of communication system with clearly marked relationships across network entities. This technique identifies faults by traversing a model of entities starting form the entity whic reported an alarm and also involves a fault identification process to identify and locate faulty network entities.

The model representation used by this technique is object-oriented representation of the given system. It is based on the OSI management framework and uses GDMO (Guidelines for Definition of Managed Objects) with extensions to the MO to model dependencies and services between the entities. This approach can make automated testing possible which can test for various testing entities availability and quality of service standards.

The event correlation in model traversing techniques is event driven as when an event occurs it is being mapped to the reported entity and the managed object representing that entity. For every event the search begins from the MO reporting the event and is searched recursively forllowing all relationship links between MO's using fault localization algorithms. Fault localization algorithms used in this technique can be of various flavours. models every event as singleton classes and merges them whenever they are traced to the same node. Typical proerties of such techniques involve

- Level at which a particular fault has occurred in the model.
- Event Type.
- Event severity.
- Event Origin - to identify if the event occurred is primary or secondary.

**Figure 3. Architecture of a model traversing technique prototype**




The MO's provide functions which lets the process test the entity for its operational status. The root cause is found when the process stops at an entity and validates that it is a malfucntioning object and doesn't depend on any other object. In a multi layer model vertical search is performed first and then horizontal search in the next lower layer to check its peer at the end of search process votes are collected to identify the faulty elements and the device that recieves the most votes is declared faulty and a testing process kicks in to test the validity of the hypothesis.

Model traversing techniques are pretty robust agianst network configuration changes and very useful in the scenario's where automated testing is a requirement and the depedency model is very natural as they are modeled as Trees or Graphs.

The biggest drawback is the Fault Propagation Patterns and when there fault is a logical combination of multiple devices and byzatine problems. It also incurs heavy testing costs.

## 4 Graph-theoretic techniques

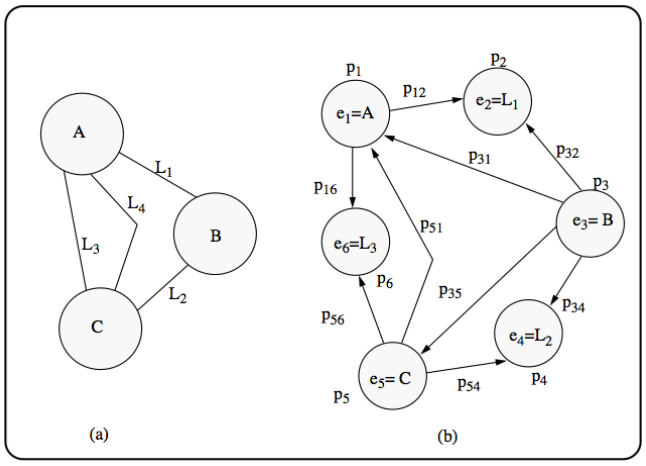
Grap theoretic techniques depend on gaphical model of a system called fault proppagation model(FPM). FPM represents fault and symptoms that occur in a system. The symptoms that are observed are mapped into nodes in the FPM and localization algorithms analyze the graph to infer the best explanation of the observed symptoms.

These techniques require a knowlede of depedencies among the abstract and physical system components and how these failures in one component are related to other components. The success of the fault localiztion algorithm depends on the the accuracy of this a priori specification.

FPM's are genrally modeled as (i) Cuasaultiy graph is is a DAG whose nodes are events and edges represent the causality implications i.e cause effect relationships between events. They are carry probabilties to the nodes and edges implying the probability of a certain event to happen. Dependency Graph is a directed graph $G=(O,D)$ where O represent a finite set of nodes and D are the depedency edges between Objects. The directed edge $(o_i,o_j) \in D$ represents that when $o_i$ fails $o_j$ fails or emits some kind of error. The edges are

also labeled with conditional probabilities. Figure 4 represents such a depedency graph.

**Figure 4. A Network example along with the dependecy graph [24]**




A lot of approaches using dependency graph, a critical assumption involves that an entity may fail in only one way. If that is the case the causal graph and dependecy graph would be the same. In the case of multiple failures associated to a single entity they can be divided into sets such as *complete failure, abnormal transmission delay, high packet loss, etc.*. The dependecy graph in this case would have multiple edges between failure modes and the Object(entity). Sethi et all in [23], [22], [21] have modeled using multiple failure modes principle.

A fault localalization algorithm, based on the provided FPM generally returns a number of hypothesis that best explain the observed symptoms. The problem of finding best explanation of the recieved alarms is *NP-hard* problem and proved in [24] by reducing the given problem to a Min Set-Cover problem.

## 4.1 Divide and conquer algorithm

The divide and conquer algorith uses a FPM assuming that one type of failure is allowed per object. It is a window based technique i.e gathers all the faults in a time window and analyzes them.

System model is a directed graph G = ( E ,D) called the dependencygraph, where E is a finite nonempty set of active terminal objects $e_i$, and a directed edge ($e_i$, $e_j$) $\in$ D denotes the fact that a fault in $e_i$ has a side effect on $e_j$ i.e., $e_j$ is dependent on $e_i$. Each vertex $e_i$ is assigned a weight $p_i$ that is the probability that the object $e_i$ fails independently of the state (failed or not failed) of any other active terminal object that is dependent on it. Each directed edge ( $e_j$,$e_i$) is assigned a weight $p_{ji}$ which represents the strength of dependency between the entities it connects. Specifically, $p_{ji}$ , is the conditional probability $p_{ji}$ = P($e_j$ fails|$e_i$ fails) that entity $e_j$, fails as a result of the failure of entity $e_i$. In addition, the weight of a directed path is defined as the product of all the weights of the edges that compose the path. Finally, the strength of dependency $p_{kl}$ between two non-adjacent vertices $e_k$ and $e_l$, is defined as the maximum of the weights of all directed paths from $e_k$ to $e_l$.

Let D($a_i$) for every $a_i$ alarm be the domain of alarms that defines as the set of objects that might have caused the alarm. The domain of an alarm calculation is a variation of single source problem that finds the min cost paths from a source vertex to all other graphs and the nodes are chosen such that the cost of path $e_i$ to $e_j$ is subject to restriction that we include only the minimum cost paths with cost $\leq$ log(W) where W is a parameter.

The outline of the algorithm run has been given below.

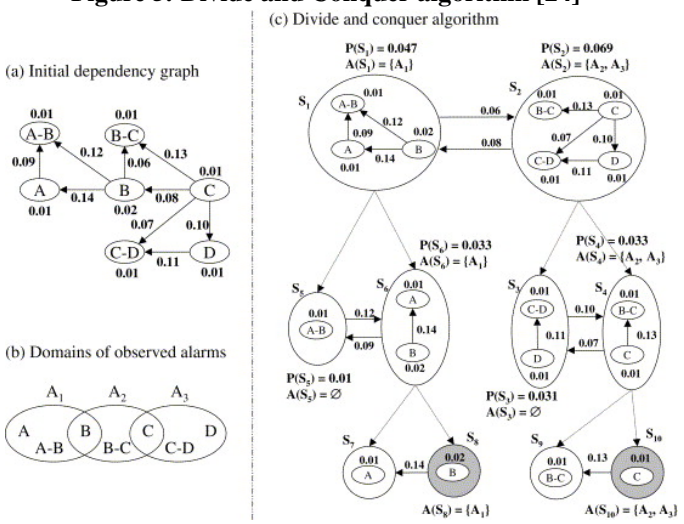
- Start with the set S of objects associated with the alarm cluster.
- Find the partitioning of the set S resulting in two disjoint sets so in each set the objects exhibit the maximum mutual dependency, i.e., for every object $e_i$ and $e_j$, that belongs to the same set and $e_k$ that does not belong to the set, the dependency weight $p_{ij}$is greater than the dependency weights $p_{ik}$, $p_{ki}$, $p_{jk}$, $p_{kj}$.
- Of the two sets, select the one that explains all the received alarms and has the maximum probability that at least one of the entities in the set is a fault. If there is no such set then find the subset of alarms that each set explains and select both.
- Apply the previous two steps of the algorithm recursively to the selected sets until the resulting sets of the current partitioning are singletons.

By partitioning with the maximum mutual dependency criterion we group together objects that are highly dependent on each other and when the result of partitionining, both the sets are capable of explaingin all the alarms then we select one set that has higher probability of having one of them at fault.

When determining atleast on memeber of the subset is faulty it is calcualted using.

$$P(S) = \sum_{\forall i: e_i \in S} [p_i + \sum_{\forall j: e_j \in S, j \neq i} p_{ij}.p_j]$$

**Figure 5. Divide and Conquer algorithm [24]**



In algorithm runs in two phases where in the first phase algorithm finds the maximum mutual partitionings of the entities in the set S. The result of Phase I is a hierarchical clustering of the entities in the set S. In the Phase II a fault localization algorithm is recursively invoked to find the subsets that are supposed to be used. The first subcluster contains all alarms that may be explained using the subset with the higher probability that one of its members was a primary source of a failure. The second subcluster contains the remaining alarms from the original alarm cluster. Its domain is the other of the two subsets. The recursive procedure is then invoked twice, taking the two subclusters and their respective domains as parameters. The recursion continues until the input alarm cluster domain is a singleton.

The algorithm mentioned above gives the explanation of the observed alarms but may not prouce the best output as it is a approximation algorithm based on maximum mutual dependency heuristic. The runtime complexity of the algorithm is $O(N^3)$ and the partitioning phase runtime is O(logN). The drawback of the algorithm is it cannot handle the lost or spurious alarms and is a time window based determinsitic algorithm.

## 4.2 Bayesian Reasoning approach

# 5 Evaluation

My evaluation criterion for the project depends on the goals I desired to achieve. The goal for me is to achieve A in this course, upon the delivery of the Mobile App communicating with cloud infrastructure and rendering of the data on Google Maps. This is the brunt of the project and requires significant effort. Enabling social integration and kalman filtering are additional characteristics. Overall I am happy with the learning outcome from this project as I learned both Android, IOS Systems and Map API's.

# A References


[1] I. Katzela Fault diagnosis in telecommunications networks, Ph.D. Thesis. School of Arts and Sciences, Columbia University, New York, 1996.

[2] I. Katzela, A.T. Bouloutas, S.B. Calo Centralized vs distributed fault localization. A.S. Sethi, F. Faure-Vincent, Y. Raynaud (Eds.), Integrated Network Management IV, Chapman and Hall, London (1995), pp. 250263.

[3] S.A. Yemini, S. Kliger, E. Mozes, Y. Yemini, D. Ohsie. High speed and robust event correlation. IEEE Communications Magazine, 34 (5) (1996), pp. 8290.

[4] Peng Wu, Rajiv Bhatnagar, Lennie Epshtein, Malini Bhandaru, Zhongwen Shi. Alarm Correlation Engine (ACE). GTE Laboratories Incorporated.

[5] G. Liu, A.K. Mok, E.J. Yang. Composite events for network event correlation - JECTOR. Integrated Network Management VI, IEEE (1999), pp. 247260.

[6] Y.A. Nygate. ECXPERT: Event correlation using rule and object based techniques. Integrated Network Management IV, Chapman and Hall, London (1995), pp. 278289.

[7] K.-W.E. Lor. A network diagnostic expert system for Acculink multiplexers based on a general network diagnostic scheme. Integrated Network Management III, North-Holland, Amsterdam (1993), pp. 659669.

[8] L. Lewis. A case-based reasoning approach to the resolution of faults in communications networks. Integrated Network Management III, North-Holland, Amsterdam (1993).

[9] R.D. Gardner, D.A. Harle. Methods and systems for alarm correlation. Proc. of GLOBECOM, London, UK, November (1996), pp. 136140.

[10] R.D. Gardner, D.A. Harle. Alarm correlation and network fault resolution using the Kohonen self-organizing map. Proc. of IEEE GLOBECOM, Toronto, Canada, September (1997).

[11] R.D. Gardner, D.A. Harle. Pattern discovery and specification techniques for alarm correlation. NOMS98, Proc. Network Operation and Management Symposium, New Orleans, LA (1998), pp. 713722.

[12] G.D. Rodosek, T. Kaiser. Intelligent assistant: User-guided fault localization. Ninth Intl Workshop on Distributed Systems: Operations and Management, University of Delaware, Newark, DE, October (1998), pp. 119129.

[13] S. Russell. Machine learning. Artificial Intelligence, Handbook of Perception and Cognition (second ed.), Academic Press, New York (1996), pp. 89133 (Chapter 4).

[14] Daphne Koller, Nir Friedman Probabilistic Graphical Models. MIT Press.

[15] M. Steinder and A.S Sethi. The present and future of event correlation: A need for end to end service fault localization. In N. Callaos et al. World Multi-Conf. Systemics, Cybernetics and Informatics, Vol XII, Orlando, FL, 2001. pp. 124-129.

[16] S. Ktker. A modeling framework for integrated distributed systems fault management. In C. Popien (Ed.), Proc. IFIP/IEEE Internat. Conference on Distributed Platforms, pp. 187198, Dresden, Germany, 1996.

[17] S. Ktker, K. Geihs. A generic model for fault isolation in integrated management systems. Journal of Network and Systems Management, 5 (2) (1997), pp. 109130.

[18] S. Ktker, M. Paterok. Fault isolation and event correlation for integrated fault management. Integrated Network Management V, Chapman and Hall, London (1997), pp. 583596.

[19] B. Gruschke. Integrated event management: Event correlation using dependency graphs. A.S. Sethi (Ed.), Ninth Internat. Workshop on Distributed Systems: Operations and Management, University of Delaware, Newark, DE, October (1998) , pp. 130141.

[20] J.F. Jordaan, M.E. Paterok. Event correlation in heterogeneous networks using the OSI management framework. H.G. Hegering, Y. Yemini (Eds.), Integrated Network Management III, North-Holland, Amsterdam

(1993), pp. 683695.

[21] M. Steinder, A.S. Sethi. Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. Comput. Networks, 45 (2004), pp. 537562.

[22] M. Steinder, A.S. Sethi. Non-deterministic fault localization in communication systems using belief networks. IEEE/ACM Transactions on Networking (2004) in press.

[23] M. Steinder, A.S. Sethi. End-to-end service failure diagnosis using belief networks. R. Stadler, M. Ulema (Eds.), Proc. Network Operation and Management Symposium, Florence, Italy, April (2002), pp. 375390.

[24] I. Katzela, M. Schwartz. Schemes for fault identification in communication networks. IEEE/ACM Transactions on Networking, 3 (6) (1995), pp. 733764.

[25] Judea Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference Morgan Kaufmann Publishers, San Mateo, CA (1988).

[26] Cynthia S. Hood, Chuanyi Ji. Proactive Network-Fault Detection. IEEE TRANSACTIONS ON RELIABILITY, VOL. 46, NO. 3, 1997.