Title
Integrated Traffic and Communications Modeling Environment for ATMIS

Authors
Varaiya, P.
Walrand, J.
Wu, F.F.
et al.

Publication Date
1997-07-01

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Integrated Traffic and Communications Modeling Environment for ATMIS

P. Varaiya, J. Walrand, F.F. Wu,
A. Polydoros, J. Sylvester

California PATH Working Paper
UCB-ITS-PWP-97-21

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department Transportation, Federal Highway Administration.

July 1997

ISSN 1055-1417

Integrated Traffic and Communications Modeling Environment for ATMIS:

Final Report for MOU 141

P. Varaiya, J. Walrand and F.F. Wu
Department of Electrical Engineering and Computer Science
University of California, Berkeley CA 94720
and
A. Polydoros and J. Sylvester
Department of Electrical Engineering
University of Southern California, Los Angeles, CA

Abstract

The aims of this **MOU** were to create an integrated environment for modeling traffic flow, sensor data, and the communications infrastructure for ATMIS. The research plan was to build separate tools for measuring the communications requirements of ATMIS traffic, modeling a mobile radio communications environment, and communications network design, and then to integrate them in an object-oriented environment from which one could also access existing or new road traffic simulation packages.

Section **1** of the report presents a framework for estimating the vehicle-roadside communications requirements taking the San Francisco East Bay region as a case study and for determining whether existing wide-area wireless technologies can be used to support those requirements. The study should be viewed within the contexts provided by the National ITS Architecture and the California ITS System Architecture **as** presented in **Caltrans'** Advanced Transportation Systems Program Plan. When completed, the study can serve **as** an example to follow for developing the communications needs for ITS deployment in California metropolitan regions.

The second **part** of the report is prepared through a sub-contract to the University of Southern California and presented as a separate volume. It provides a modeling framework for the mobile radio communications environment.

Section **2** of the report describes enhancements made to the NetPlan planning tool. NetPlan is a tool for designing wireline communication networks. It has models of LANs (eg. Ethernet) and WANs; standard communication protocols (eg. CSMA/CD, TCP/IP); and various simulation and network optimization tools. The NetPlan enhancements made under this project deal with incorporating ATM networks in NetPlan libraries. The detailed results appear in a PhD thesis.

For reasons outlined in Section 3, the aim of integrating these tools in an object-oriented environment that could access traffic simulation packages was abandoned. In its place we have developed the framework presented in Section 1.

Executive Summary

The original aims of this MOU were to create an integrated environment for modeling traffic flow, sensor data, and the communications infrastructure for ATMIS. The plan was to build separate tools for measuring the communications requirements of ATMIS traffic, modeling a mobile radio communications environment, and communications network design, and then to integrate them in an object-oriented environment from which one could also access existing or new road traffic simulation packages.

The first section of the report presents a framework for estimating the vehicle-roadside communications requirements taking the San Francisco East Bay region as a case study and for determining whether existing wide-area wireless technologies can be used to support those requirements. The study should be viewed within the context provided by the National **ITS** Architecture [1-4] and the California **ITS** System Architecture as presented in Caltrans' Advanced Transportation Systems Program Plan [9]. The study can thus serve as an example to follow for developing the communications needs for ITS deployment in California metropolitan regions. The National **ITS** Architecture study illustrated how an **ITS** architecture might be deployed in a hypothetical urban area called "Urbansville" [5]. Our study is motivated by this study of Urbansville in the more concrete setting of the San Francisco East Bay region. Our study offers two lessons: first, it shows how one can systematically ***go*** about evaluating **ITS** deployment requirements; second, in the absence of data about the market for ITS services, one is forced to make many educated guesses. Over time, as ITS services become available, such guesses will be replaced by more reliable estimates. The complete study will appear later this year as a doctoral dissertation [12].

The second section of the report is prepared by University of Southern California and presented **as** a separate volume. It provides a modeling framework for the mobile radio communications environment. The emphasis here is on the physical channels. This section is a continuation of earlier work [10,111

The third section of the report summarizes enhancements made to the NetPlan planning tool [13]. NetPlan is a tool for designing wireline communication networks. It has models of LANs (eg. Ethernet) and WANs; standard communication protocols (eg. CSMA/CD, TCP/IP); and various simulation and network optimization tools. The NetPlan enhancements made under this project deal with incorporating **ATM** networks in NetPlan libraries. The detailed results appear in a PhD thesis [14]. The earlier effort in developing NetPlan was funded by Pacific Gas & Electric Company.

As mentioned above, it was an objective of the original proposal to build an object-oriented environment that would incorporate these planning tools and provide access to existing or new traffic simulation packages. The aim was to permit a design team to work using a common interface with needs assessment procedures (section 1), **ITS** service demand generated on the basis of traffic simulation packages, communication network design tools (sections **2** and **3).** It was thought that such a common interface could be provided within an object-oriented environment. That objective was not fully met for reasons more fully explained in section 4. The basic reason is that existing traffic simulation packages, network planning tools, and needs assessment procedures are so heterogeneous--- reflecting differences in their design, purposes, choice of programming language, etc.--- that they cannot usefully be integrated using commercially available object-oriented software systems.

Finally, in section ***5*** we evaluate our study within the context of the overall PATH program.

1. ATMIS Vehicle-Roadside Communication Needs

Tetiana Lo[1]

Abstract

This section describes a framework to determine if existing wide-area wireless technologies can be used to support vehicle-to-roadside ITS services in the San Francisco East Bay Area.

1.1. Introduction

Transportation statistics in the United States indicate that present-day traffic problems will carry on into the future and be accompanied by increasing congestion, pollution and decreased driving efficiency if current trends continue. Recent improvements in such areas as vehicle emissions and fuel efficiency, the building of new roadways and increasing the number of lanes on existing roads have alleviated the problems considerably. However, rapid growth in traffic demand and volume may overwhelm these incremental solutions. **A** fundamental shift in approach is needed to effectively resolve these complex transportation issues.

The Intelligent Transportation Systems **(ITS)** program is the most important approach that promises to achieve safer, more reliable and more economical traffic flow on the existing roadway system. Established by the Intermodal Surface Transportation Efficiency Act of 1991, **ITS** is the national initiative to apply information, communications, and control technologies to improve surface transportation efficiency and to limit the negative effects on the environment and society.

California's **ITS** program preceded the federal initiative by several years. Caltrans was an active participant in the shaping of the National ITS Architecture in order to ensure that California's transportation problems were addressed, to influence the architecture design ***so*** that it supports the full utilization of California's existing transportation infrastructure, and to deploy **ITS** systems in the future that can fit well within the emerging national **ITS** architecture [9, p. 20].

An early outcome of the national planning process was the identification of a number of capabilities---"user services"---which, if deployed, would collectively meet the goals of **ITS.** Most of these services require the flow of data or voice messaging between vehicles, remote sites, and coordinated control centers; hence, the need for extensive wireless communications. Until recently, wireless communications alternatives had not been studied in detail with respect to **ITS** needs. In this work we critically examine the current state of the **art** and present a framework to evaluate how well these alternatives satisfy the **ITS** requirements.

ITS architecture development activities in the area of communications have encompassed two main thrusts: (1) communication architecture definition, i.e., selection of communication service and media types to interconnect the appropriate entities engaged in transportation decisions, and (2) several types of inter-related communications analyses to ensure the feasibility and soundness of the architectural choices made in the definition. Thus far, efforts have focused primarily on the development of **a** strategy for the phased deployment of the communications architecture to support **ITS** user services for several government-specified scenarios and time frames. Practically speaking, these strategies cannot be directly applied to most potential ITS service areas. System design decisions made for an ITS deployment region are heavily dependent on numerous factors particular to that region, such as physical terrain, user service pref-

---

[1] Department **of** Electrical Engineering and Computer Sciences, U.C. Berkeley.

erences, and technological availability. The ITS service provider must be capable of adapting the system architecture design to the needs and features of the service area and hence, customers, under consideration.

The objective of this work is to develop a blueprint for ITS service providers contemplating the deployment of ITS user services requiring wide-area wireless communications. This framework is based upon the basic structure and user services defined by the National ITS Architecture [1]. In particular we examine the San Francisco East Bay Area and through simulations determine quantitatively if an existing wireless technology, Cellular Digital Packet Data, can be used to support the ITS user service requirements established for our scenario. The report is organized as follows. We first present a general overview of the National ITS Architecture including a discussion of the user services, the subsystem components, and the data flows that must be supported. We then describe our scenario of interest, the San Francisco East Bay Area, and discuss the selection of ITS user services appropriate for this region. An analysis of the ITS data loading requirements follows. The latter half of the study focuses on wireless wide-area communications alternatives to support vehicle-to-roadside user services. We present an in-depth review of candidate technologies and propose performance evaluation criteria to allow the ITS system designer to more effectively compare communications infrastructure alternatives. The wireless system simulation results based on our scenario are presented. Lastly we discuss implementation and deployment issues.

1.2. The National ITS Communication Architecture

In general, the Communication Architecture for ITS [2] has two components: one wireless and one wireline. The wireless component provides tetherless users, usually in vehicles, with the means to exchange information with one another, as well as to obtain access to geographically fixednetwork resources. Based even on a cursory examination of the ITS user requirements, it is clear that a diverse range of communication requirements must be supported. The National ITS Architecture has proposed to divide the wireless portion into three parts:

- Wide-area wireless infrastructure, supporting wide-area information transfer. For example, the direct use of existing and emerging mobile wireless systems.
- Short-range wireless infrastructure for short-range information transfer (limited to specific applications), characteristic of transmissions between a mobile user and a base station, eg. systems used for electronic toll collection.
- Dedicated wireless system handling high data rate, low probability of error, fairly short range, Advanced Highway Systems related data flows, such as vehicle-to-vehicle transceiver radio systems.

In order to determine the specific communication needs of the ITS network, the data flow requirements for interconnections between the transportation systems and subsystems, eg. connection between an Information Service Provider (ISP) Subsystem and a Vehicle Subsystem (VS), must be provided. These data flows are then mapped to the communication service requirements (which are generic information exchange capabilities such as messaging data), and the required types of interfaces, i.e., wireline or wireless, associated with each flow are established. The Communication Architecture specifies the communication technology family most applicable to each data flow, eg. two-way wide-area wireless broadcast. This definition precedes identification of a specific technology or system. In practice, the final step of selecting a given technology would be performed by the ITS implementor or service provider.

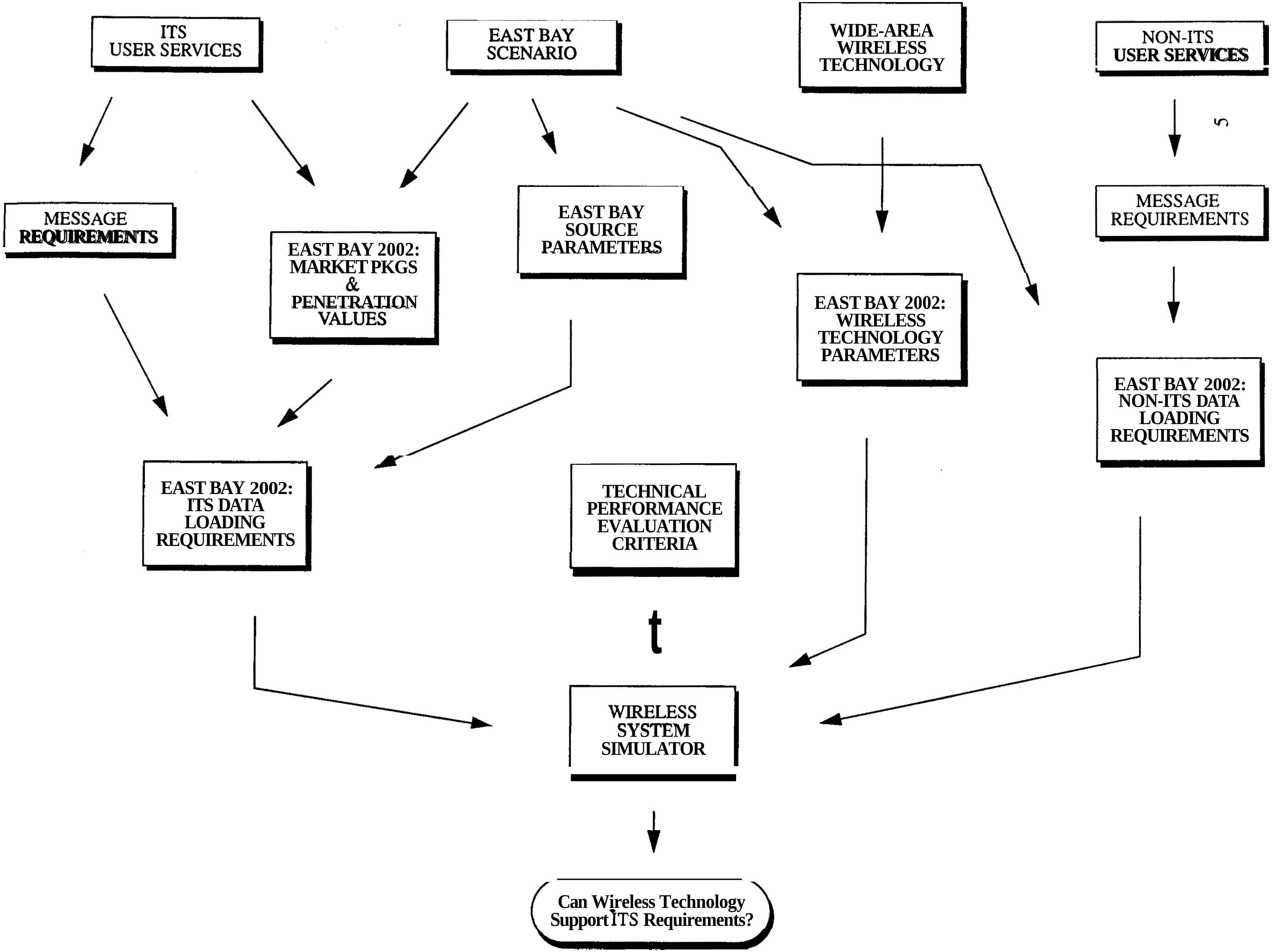

Figure 1-1 ITS Vehicle-to-Roadside Communications Architecture Design Framework

1.3. ITS Communications Architecture Design Framework

In order to accurately assess the overall end-to-end system performance, it is necessary to analyze both wireline and wireless components and the switching/routing elements of the communications infrastructure. In this study, however, we focus on the wide-area wireless portion. In Phase I of the National ITS Architecture effort, it was determined that the wireline component of the communications systems supporting ITS will not constitute the communications bottleneck. In fact, through proper design, and given the many alternatives available, the throughput of wireline systems can be made to meet the users' requirements (with any desired margins) satisfying any least cost criterion.

While the sharing of wireless resources and infrastructures between non-ITS applications and ITS service offerings is likely, in our initial investigation we will assume that the underlying communications network is solely dedicated to ITS use. Our objective is to test the feasibility of strictly utilizing a single existing wireless technology to support the desired ITS services for our East Bay scenario and in the five-year timeframe up to **2002.** After performing this preliminary examination, the non-ITS traffic can be incorporated, and its impact on the performance of the wireless technology can be assessed.

The design and performance evaluation framework for the wide-area wireless communication architecture in the San Francisco East Bay Area is illustrated in Figure 1-1. At the highest level we are concerned with three main entities: ITS User Services, the East Bay Scenario, and a Wide-Area Wireless Technology. Non-ITS applications can later be incorporated in our data loading analysis. The following sections will detail the various elements and flows of the diagram.

1.4. Data Loading Requirements for East Bay Scenario

1.4.1. Message Requirements of User Services

The ITS Architecture has defined a set of user services that represents the basic building blocks that will support the deployment of advanced ITS capabilities. Currently there are twenty-nine user services that fall under the seven general areas as shown in Table 1-1.a

The upper left block of Figure 1-1 shows the need for determining message requirements from ITS user services. A market package represents a service that will be deployed as an integrated capability. Examples include Transit Vehicle Tracking, Network Surveillance, Emergency Response, and Freight Administration. The ITS Implementation Strategy [3] illustrates the many-to-many relationship between the market packages and user services. For example, the market package Broadcast Traveler Information supports the three user services Pre-Trip Traveler Information, En-Route Driver Information, and En-Route Transit Information. Associated with each market package are various equipment packages consisting of hardware and/or software in multiple subsystems, that implement the market package functions and are likely to be purchased by the end-user. Based on these equipment packages one can determine the flows in terms of message length and frequency between the subsystem entities, from the ITS Logical Architecture [4].

1.4.2. East Bay Area Scenario

The ITS scenario we study in this report is based on two counties in the San Francisco Bay Area in California, namely, Alameda and Contra Costa counties, commonly referred to as the East Bay Area. The

Table 1-1. **ITS** User Services

Tkavel and Tkansportation Management En-route driver information Route guidance Traveler service information Traffic control Incident Management Emission testing and mitigation	Commercial Vehicle Operations Electronic Clearance Automated roadside safety inspection On-board safety Monitoring Administrative processes Hazardous material incident response Fleet management
Travel Demand Management Pretrip travel information Ride matching and reservation Demand management and operations	Emergency Management Notification and personal security Vehicle management
Public Transportation Operations Transportation management En-route transit information Personalized public transit Public travel security	Advanced Vehicle Control and Safety Systems Longitudinal collision avoidancb Lateral collision avoidance Intersection collision avoidance Vision enhancement for crash avoidance Safety readiness Precrash restraint deployment Automated highway systems
Electronic Payment Payment services	

region **is** approximately 1,458 square miles and will have a population of 2.7 million in 2010. The scenario **was** modeled through the use of the “Urban Scenario Guide, Urbansville, Phase 11” [5] and government-supplied statistical data [6].

In order to calculate the ITS data loading requirements for the East Bay scenario, the projected market packages and the associated penetration values for the region must be determined for the year 2002. The development of a viable deployment strategy is an involved and complicated task in itself. The major considerations include:

- Identification and prioritizing of transportation problems in the East Bay area.

- Definition of ITS goals and objectives for the scenario.
- Evaluation of ITS user services with respect to the effectiveness of meeting ITS goals.
- Identification of ITS technologies which support user services.
- Evaluation of ITS projects and programs.
- Investigation of funding sources, public/private partnership opportunities, and institutional considerations.

The market package deployment plan we propose is based on several sources. We focus primarily on the "ITS Early Deployment Plan," (EDP), developed by the Metropolitan Transportation Commission [7]. The EDP Action Plan defines a series of services to support implementation of eight strategies for using ITS in the Bay Area. These include deploying a probe vehicle system, transit fleet management systems, and corridor transportation management systems with the purpose of "giving travelers the information needed to make the best travel choices, and giving system operators the data they need to make the transportation system more efficient." These proposed projects will be investigated and evaluated in terms of their relevance to the East Bay Area. National and regional statistics are used to determine the East Bay source parameters, i.e., potential user population of the equipment packages (eg. number of long-haul and transit vehicles), the number of various facilities and centers (eg. commercial vehicle operations facilities and traffic management centers), and roadway characteristics (miles of freeway and arterial surface streets), Market penetration values for users of ITS will be assigned. Together with the message requirements, we will determine an estimate for the data transmission requirements for a candidate deployment of the ITS architecture in the East Bay.

1.5. Wide-Area Wireless Technologies

The vast majority of two-way wide-area wireless flows are best supported by commercially available mobile wireless data networks operating in the packet switching mode[8]. Prominent among these are Cellular Digital Packet Data (CDPD), Personal Communications Services (PCS), and private packet radio network systems such as RAM and ARDIS.

All of these technologies have the capability to meet ITS wireless communications requirements. CDPD, however, is the only one standardized with an open system architecture, and effectively the only cellular data system commercially available. CDMA data systems are evolving in the standards bodies but many of the claims for CDMA have yet to be proven in real-world applications. Private Mobile Data Networks, as well as Specialized Mobile Radio (SMR) systems will be analyzed with a level of detail warranted by their proprietary nature. ARDIS and RAM will also be assessed in terms of their ability to provide mobile solutions. Finally, two-way paging, the first service made available over the newly licensed Narrowband-PCS, is examined to assess the potential of its somewhat limited scope.

To meet the requirements imposed by the twenty-nine user services defined in the National ITS Architecture, and to provide services to a wide range of areas including rural settings, a hybrid of terrestrial and satellite services may be required in some situations. In this study we focus on terrestrial wireless data systems, but we will provide an overview of projected satellite services, with particular emphasis on low-earth-orbit (LEO) systems, which seem to the most appropriate for ITS among satellite systems since they are targeted specifically to short bursty data transactions.

1.6. Wireless System Simulation

Our evaluation strategy leverages existing communications infrastructures, i.e., cellular technologies, and utilizes developed expertise in the area of wireless network simulation. More specifically, the evaluation uses and builds upon simulation tools that have been developed for real systems that have been and continue to be deployed commercially. For practical reasons, the wireless simulation effort had to be constrained to open, already standardized systems. Thus, given that Cellular Digital Packet Data (CDPD) is the only fully standardized, open system for data communications over cellular, with the advantage of being in an advanced deployment stage, CDPD is used as a case study to show the feasibility of a wide-area **ITS** cellular solution.

In terms of technology availability, CDPD provides full coverage of the East Bay. The CDPD scenario and technology parameters necessary for network simulation include user profiles, topographical and topological parameters, and the distribution and characterization of base stations. These will be obtained from FCC filings and other government information sources.

We are currently investigating wireless system simulation tools to simulate CDPD systems using actual cellular deployment information from the East Bay Area. The leading contender **is** the MObile System Simulator **(MOSS),** a proprietary software package developed at GTE Laboratories for simulating the performance **of** mobile and wireless communications systems providing voice and data services. AMPS analog cellular systems, **IS-54 TDMA** cellular systems and CDPD packet data are amongst the models incorporated into **MOSS.**

1.7. Technical Performance Evaluation Criteria

In order to effectively evaluate the performance of the CDPD under **ITS** data loading, we propose to obtain simulation results of the following quantitative technical system performance evaluation criteria: coverage, delay, throughput, and information on a link by link level about utilization and bandwidth. By comparing these values with those imposed by service requirements for the user services and our defined scenario we will able to determine if the CDPD is capable of meeting **the** application requirements. In this study we also intend to define quality of service requirements in terms of response time (i.e., maximum tolerable delay) for the user services which will be deployed. Figure 1-2 givers **an** overview of the communication system simulation we propose.

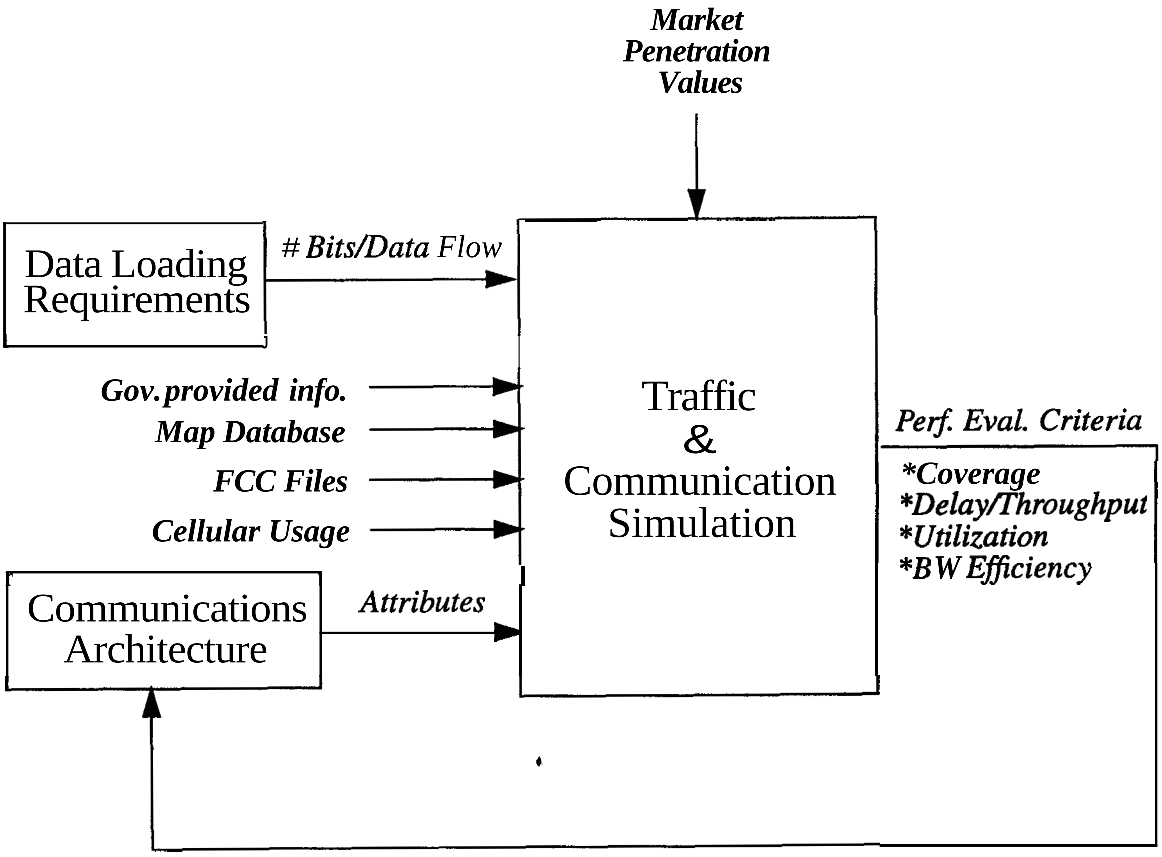

Figure 1-2 Wide-Area Wireless System Simulation

1.8. References

[1] "National ITS Architecture," June 1996, US DOT, **FHA.**

[2] "National ITS Architecture, Communications Architecture," April 1996, US DOT, FHA.

[3] "National **ITS** Architecture, Implementation Strategy," June 1996, **US** DOT, FHA.

[4] "National ITS Architecture, Logical Architecture," May 1996, **US** DOT, FHA.

[5] "Urban Scenario Guide, Urbansville, Phase 11," March 13, 1995.

[6] "San Francisco Bay Area County and Regional Profiles," September 1994, Metropolitan Transportation Commission.

[7] "MTC Completes Early Deployment Plan," Smart Moves, Issue 5, Metropolitan Transportation Commission, September 1996.

[8] 'Wireless Communications for Intelligent Transportation Systems," S.D. Elliott, D.J. Dailey, Artech House, Inc., 1995.

[9] "Advanced Transportation Systems Program Plan. 1996 Update: Framework for a California Partnership," December 1996, New Technology and Research Program, California Department of Transportation.

[10] A. Polydoros, et al. "Vehicle to Roadside Communications Study," PATH Research Report, UCB-ITS-PRR-93-04, June 1993.

[11] A. Polydoros, ***"An*** Integrated Physical/Link-Access Layer Model of Packet Radio Architectures," PATH Research Report UCB-ITS-PRR-95-33, September 1995.

[12] **T.** Lo, "A vehicle-to-roadside **ITS** study of the San Francisco East Bay area," Ph. D thesis, University of California, forthcoming, 1997.

[13] S-M. Lun, F. Wu, N. Xiao and P. Varaiya, "Netplan: An integrated network planning environment," in ***State of the Art in Performance Modeling and Simulation: Computer and Communication Networks, Vol I***, Kluwer, 1993.

[14] Z. Qin, **"ATM** network planning and management," Ph.D. thesis, University of California, 1996.

2. Designing B-ISDN Network Topologies Using The Genetic Algorithm

Zhigang Qin[1], Felix F. Wu[2], and Nathan Lad

Abstract

The topological design of B-ISDN networks is addressed. We model the topological planning **as** a non-linear mixed-integer programming problem. The genetic algorithm, an effective optimization method, is applied to this problem. Since the randomness of the genetic algorithm cannot guarantee the biconnectivity requirement in the topologies generated by the genetic algorithm, we propose an algorithm to make all topologies at least biconnected while increasing the overall cost of the topologies the least. The result for a 20-node test case is presented in the paper and it is shown that the algorithm we propose has a very good convergence property.

2.1. Introduction

B-ISDN (Broadband Integrated Services Digital Network) is expected to be the main carrier of next generation communication networks and the Asynchronous Transfer Mode (ATM) will serve as the basic **transmission** technology for B-ISDN. The B-ISDN technology and the rapidly increasing information technologies will lead to rapid increasing in traffic load and frequent shift in traffic pattern. The changing traffic pattern and the new technologies used in ATM networks make the topological design of **ATM** networks a new problem. Moreover, most research that dealt with topological design were either for data packet networks or for circuit switched networks, and most technical papers about topological design were published during the late 1960s and early 1970s [1]. To our knowledge, there is no technical paper concerned with topology planning for ATM networks. Therefore, we extend our previous work on network planning of packet networks [2] to ATM networks.

Because of the rapid expansion in both circuit-switched and data packet networks, topological planning has been **an** active research topic. However, there is no closed-form algorithms for optimizing a network topology for circuit-switched or data applications. The designer must employ a combination of heuristic algorithms and analyses in an interactive manner until a satisfactory solution is obtained. In [1], network planning methods which rely on human interventions are summarized. One method emphasizes expert experience and tries to find a satisfactory solution using a specific topology like **star,** ring, etc. Heuristic algorithms that are not based on a specific topology were also proposed. Perturbation techniques are the main approaches. The most important perturbation techniques are the branch exchange method, cut-saturation method, and concave branch elimination method [10]. Of these three types of heuristic algorithms, cut-saturation algorithms gives the better results and is computationally more efficient than the other two. In practice, these algorithms are usually combined with other heuristic algorithms or some variations are used to improve result and computational efficiency. The main drawback of these methods **is** the likelihood **of** becoming trapped in a local optimum at an early stage. And there are indeed many local optima for the topology design problem [11]. Algorithms based on integer and nonlinear programming techniques were also proposed [3][4]. The problem with this kind of technique is that it is difficult to deal with large networks and simplifications or assumptions have to be made to solve them. Thus, they lack flexibility and are not realistic in practice.

Because **of** the drawbacks in the existing algorithms, we propose to use a different optimization technique, genetic algorithm, which can take the advantage of the layered architecture **of** ATM networks and may find

---

[1] Optimal Networks, formerly with Department of Electrical Engineering and Computer Sciences, **U.C.,** Berkeley.

[2] Department **of** Electrical Engineering **and** Computer Sciences, U.C., Berkeley, currently on leave with the University ***of*** Hong Kong.

[3] University **of** Hong Kong.

the global optimum, to solve the problem of ATM network planning. In subsequent sections of this paper, the details of this approach is discussed. Section 2.2 presents the model for the ATM network planning. Section 2.3 describes the algorithm for solving the problem. In Section 2.4, results are presented for a 20-node ATM network. Conclusions and suggestions for future work are given in Section 2.5.

2.2. Mathematical Model for ATM Network Planning

ATM is a layered architecture. Three layers have been defined to implement the features of ATM. The top layer is the ATM Adaptation Layer (AAL). The second layer is the ATM layer. The bottom layer is the physical layer. The ATM connection is set up in the following way: when information needs to be communicated, the sender negotiates a "requested path" with the network for a connection to the destination. When setting up this connection, the sender specifies the type, burstiness, other attributes of the call, and the requirements of the end-to-end quality of service. The network checks its own resources to see whether such a quality of specification can be guaranteed. If yes, a virtual channel connection will be set up, otherwise, the request will be rejected. To accurately model those characteristics of ATM networks, we propose to use a layered framework for network planning and management. In the following paragraphs, we describe the framework in detail.

The planning and management task is divided into four levels. The first level is the physical network design, which is performed to accommodate traffic over a relatively long period, e.g., one year or a few years. The traffic requirements at this level are merely an estimate. Thus, complete characterization of the traffic dynamics complicates the problem and is unnecessary. The **task** at this level is to design the network topology and assign physical link capacities based on the estimated long-term traffic requirements. The objective of topology design is to minimize the total cost. The second level is the design of the logical network, i.e., the reconfiguration of the network, given the physical network and a more detailed traffic description. The task at this level is to determine the virtual paths and assign to them their bandwidths. The third level is the call control level where admission control takes place. At this level, a more detailed description of the incoming call is given and the admission control functions decide whether to accept the call based on the resources available at the moment. The fourth level is the cell level. The main **task** here is to police the traffic so that users will not send more traffic than agreed upon by the network and the users.

The layered model simplifies the task of the topology planning, since the ***QOS*** requirements are moved down to the lower levels. According to the layered model, the topological design problem can be stated as follows:

Given:

- node locations
- traffic requirements
- link cost matrix (fixed and variable costs)
- nodal costs

Objective:

Select the links and their capacities to minimize the total cost.

Subject to:

- traffic flow constraints

- reliability constraints
- traffic requirements

A mathematical formulation for the topological design problem is given in following:

Variable definition:

sd source destination pair
D: set of destination hosts
N: set of all nodes
S set of source hosts

$$x_{ij}^{sd} = \begin{cases} 0 & \text{pair } sd \text{ does not choose link } ij \\ 1 & \text{pair } sd \text{ chooses link } ij \end{cases}$$

y_{ij}^{sd} : bandwidth assigned to link ij for pair sd;
t^{sd}: the average traffic requirements for pair sd;
c_{ij}: variable cost for link ij (per unit bandwidth);
c_{ij}^{f} : fixed cost for link ij;
c_{i}^{n} : nodal cost for node i (per unit capacity);

Mathematical formulation:

$$\min{}' \sum_{j}\sum_{sd}(c_{ij}y_{ij}^{sd} + c_{i}^{n}y_{ij}^{sd} + x_{ij}^{sd}c_{ij}^{f}) \tag{1}$$

S.T.

$$y_{sj}^{sd} \geq t^{sd} \qquad s \in S, d \in D \tag{2}$$

$$y_{id}^{sd} \geq t^{sd} \qquad s \in S, d \in D \tag{3}$$

$${}'^{sd}_{ij} = \sum_{j} y_{ji}^{sd} \quad i \in N, i \notin S, i \notin D, s \in S, d \in D \tag{4}$$

$${}^{d} \geq 0 \qquad i, j \in N, s \in S, d \in D \tag{5}$$

$${}^{t} = 0 \text{ if } y_{ij}^{sd} = 0 \quad i, j \in N;\ s \in S; d \in D \tag{6}$$

The objective function (1) is to minimize the total cost, which consists of link cost and nodal cost. The link cost consists of two parts: the fixed cost and the variable cost. The fixed cost of a link does not depend on the capacity of the link, and only depends upon the decision of whether to set up a link there. This could be interpreted as the infrastructure cost of building a link between two nodes. Since the variable x_{ij}^{sd} can only take either 1 or 0 values, the summation of x_{ij}^{sd} will either be 1 or 0, even if multiple routes choose the same link. Thus the fixed cost will not be counted multiple times. The variable cost of a link is proportional to the capacity of the link. So, the link cost is modeled as a piecewise linear function. It should be pointed out that the algorithm we propose has the ability to handle any functions, even functions which do not have close-form formulations. Equations (2) and **(3)** specify that all traffic requirements must be satisfied. Since we are in the planning stage, the available link capacity may not be the exact amount of the traffic requirement. Therefore, the link capacity could be greater than the traffic requirement. Equation (4) specifies that the traffic flow is conservative, *i.e.*, for any intermediate node, the traffic that goes in the node should equal to the traffic that comes out of the node. Equation *(5)* states that the link bandwidths cannot be negative. Equation (6) associates the decision variable x_{ij}^{sd} with the bandwidth variable y_{ij}^{sd} .

The above formulation is a non-linear mixed-integer programming problem, which is very difficult to solve. In following section, we propose to use the genetic algorithm to solve this problem.

2.3. Algorithm

The characteristic difficulty of the topological design problem formulated above is that the number of constraint equations quickly becomes unmanageable for even a small problem, if a conventional solving technique is used. If a heuristic approach, such as the perturbation algorithms, is used to solve the problem, only the local optima can be found. Thus, a different, stochastic optimization technique, simulated annealing algorithm, is proposed for topology planning of packet-switched networks in [8] and is shown to be successful. The advantage of a stochastic search algorithm is the increased likelihood of finding the global optimum **and** the applicability to various problems. Therefore, in this research, we propose to applf a stochastic algorithm, genetic algorithms, to study the problem, because genetic algorithms have a better chance than simulated annealing of finding the globally optimal solution [6]. The disadvantage of genetic algorithms is that they may be computationally more intensive than simulated annealing.

A genetic algorithm consists of three main procedures: the determination of the initial solution pool, the selection of parents, and the production of offspring. The generation of initial solution pool **is** to create a population of chromosomes, which represent feasible solutions to the original problem. A good initial solution pool should be randomly selected. After the initial solution pool is selected, some solutions are selected based on the evaluation of the each individual solution (fitness) and the outcome of random choices as the parents of next offspring. The next step is the production of offspring. The production of new offspring typically undergoes crossover and mutation. Crossover combines and mixes different individuals being selected to form new ones. Mutation is performed on each individual which changes each gene of the individual with a small probability. After these procedures, we get a new population. The new population will display patterns of behavior that are more like those of the successful individuals **of** the previous generation, and less like those of the unsuccessful ones. With each new generation, the individuals with relatively better values will be more likely to pass on to the next generation, while the relatively unsuccessful individuals will be less likely to pass on [7]. The solutions are improved by going through a large number of generations. In 1994, Rudolph Guenter proved that the simple genetic algorithm converges to the global optimum [5].

The difficulty in developing a genetic algorithm to solve a particular optimization problem lies in the necessity of developing appropriate representation and encoding scheme for the solution space. The performance of a genetic algorithm heavily depends on solution representation, encoding scheme, and selection **of** genetic operators.

For the ATM network planning problem, we propose to obtain an initial solution pool by the following procedure. 1) Specify a value k which is the degree of the network that we are going to generate. Starting from node 1 to node n (the total number of nodes), for every node x, determine the number of links l that are incident upon this node. Assume l is less than k, determine the k-l neighboring nodes $y_1, \ldots y_{k-l}$ that have the least link costs (variable cost) with node x and do not have a connection with x. Make connections between x and $y_1, \ldots y_{k-l}$. When we finish checking all the nodes, a k-degree network topology is obtained. Thus m initial network topologies, which make up the initial solution pool, can be obtained by giving m different values to k. **2)** Check the connectivity of these network topologies. If any of the topologies is not a connected graph, we add the least-cost link which connects the disjoint components of the topology to make it connected.

After the initial topologies are generated, the next step is to assign capacities to links. This is essentially a routing problem. In our study, the shortest path routing in terms of the link cost is used. We proceed **as** follows: for all source-destination pairs, one at a time, we find the shortest path between the source and the destination, and increase the capacities of all the links along the path with the traffic requirement of this source-destination pair. After the capacity assignment **is** finished, an initial solution pool is established.

Next we evaluate the solutions generated and select candidates for producing offspring. The following fitness function is proposed for the evaluation:

$$f_i = \frac{1/c_i}{\sum_j 1/c_j} \tag{7}$$

where c_i is the cost of *i-th* solution in the solution pool. The parents are selected according to the fitness of **each individual, *i.e.*, the probability** of **each individual being chosen is the fitness value. To make the algorithm converge, we** force **the selection** to **include the best solution so far.**

The offspring is produced by crossover and mutation. For crossover and mutation operations, a representation **of** the topology and an encoding scheme are needed. We represent a topology with a *0-1* matrix *topo*. The matrix element *topo(i,j)* equals 1 if there is a link between node i and node j, and equals 0 if there is no link between them. Because the matrix is symmetric, we only need the upper-triangle elements to represent the topology (diagonal elements excluded). Therefore, the topology can be represented by a binary triangle matrix. Since the crossover and mutation operators need a binary string representation, we combine the upper-triangle elements to form a binary string with $n(n-1)/2$ elements, which will undergo the crossover and the mutation. The crossover operator we adopted interchanges the elements of the two

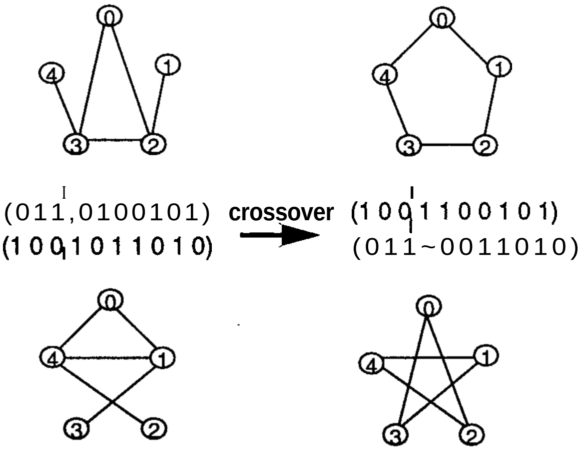

Fig. 1. Illustration of Crossover Operator

topology strings up to a point which is randomly determined. This process is shown in Fig. 1. After the crossover operations, the mutation operator is applied to all newly generated topology strings. The mutation operator randomly selects a small portion of the elements in the selected candidates and changes the value of the elements to the reverse value, for example, from 1 to 0 or from 0 to 1. In other words, the mutation operator takes off or adds some links randomly from some offspring topologies. The percentage of links which go under mutation operation should be less than 3%. In fact, crossover and mutation are very similar to perturbation methods in that both search for the best solution by adding and removing links. The genetic algorithm searches for a better solution by increasing the chance for good solutions to be selected and tries to combine the good parts of two solutions to form an even better solutions through crossover operations. The mutation operations try to lead the search out of local optima or give a new start point for searching. The advantage of the genetic algorithm over the perturbation methods is that it works on a pool of solutions instead of one, and the adding or removing of links is not limited to those operations that necessarily improve the objective value, so the algorithm has a better chance of escaping local minima [6].

After the crossover and the mutation operations, we assign capacities to the links of all topologies using the same shortest path routing algorithm as the one used for generating the initial solution pool. A new generation of solutions is produced when the capacity assignment is finished.

One important point is that a disconnected topology may be produced by the crossover and the mutation. Adding back links is crucial to the algorithm for maintaining feasibility of offsprings and reducing their costs. The proposed approach is to add back the minimum cost links which connects the disjoint components. The procedure is as follows: assume that a disconnected topology is generated by the crossover and the mutation operations. First, we select a source-destination pair which has a source in one subnetwork and a destination in the other subnetwork. Then we use Dijkstra's algorithm [9] to generate two minimum-cost spanning trees rooted from the source and the destination of this source-destination pair for both disconnected subnetworks. Then we compare the costs of all links that connect the leaf nodes of these two disjoint components, and add the link of the least cost to the topology. This process is repeated for all other source-destination pairs which have a source in one subnetwork and a destination in the other subnetwork without considering the newly added links (otherwise, the network is already connected and there is no need to add a link to the topology). After this operation, the topology will become connected again.

The above algorithm does not consider the reliability constraint. For network in which the reliability is important, biconnectivity (2-connectivity) or n-connectivity constraints are required. In practice, biconnectivity is normally sufficient for reliability. Moreover, higher connectivity implies more redundant resources. Thus, only biconnectivity constraints are considered in our algorithm. Before we explain the algorithm for converting a non-biconnected graph to a biconnected graph, we introduce a new concept, a lemma and a theorem.

Definition. A biconnected component is called peripheral if it has only one articulation point.

Lemma. Let ***G*** be a graph, G1 and ***G2*** be two peripheral biconnected components of the graph G, and G3 be the smallest subgraph of G which contains G1 and G2. If a link is added between any nodes of these two components except for the articulation points of G3, then resulting subgraph **G3** is biconnected. For all such links, the link which has the minimum cost is called the bridge link.

Proof. Before the link is added, the only node whose removal will split the graph is the articulation point. After this link is added, this is no longer true.

Theorem. Let *n* be the number of peripheral biconnected components of a non-biconnected graph. Then the minimum number of links needed to make it biconnected is *n-I*. The minimum cost to make it biconnected is the cost of the *n-I* bridge links.

Proof. By the lemma above, we need at least one link to make a graph with two peripheral biconnected components biconnected and the bridge link is the least cost link of all such links. Adding such a link also reduces the number of peripheral biconnected components by **1.** Thus, for a graph with *n* peripheral biconnected components, by induction, we need at least *n-I* links to make the graph biconnected and the minimum cost is the cost of the *n-I* bridge links.

Based on the discussion above, we propose an algorithm for **ATM** network planning with biconnectivity constraint. The block diagram of the algorithm is shown in Fig. 2. In the algorithm, the biconnectivity constraint is checked after a network topology is generated. If the topology is not biconnected, a bridge link is added to the topology to make it biconnected. The link capacities are assigned according to the shortest path routing algorithm which increases the capacities of the links along the routes by the amount of traffic requirements. For the links which are not among the shortest paths, there are two ways to assign the capacities. The first way is to assign the minimum link capacity given by users. The second way is to assign the minimum alternative path capacity. The minimum alternative path capacity is defined as follows: assume that concatenated links *i,...,**j*** are not among any of the shortest paths and nodes *i* and j are the terminating nodes of these links. The minimum alternative path capacity is the minimum path capacity of all alterna-

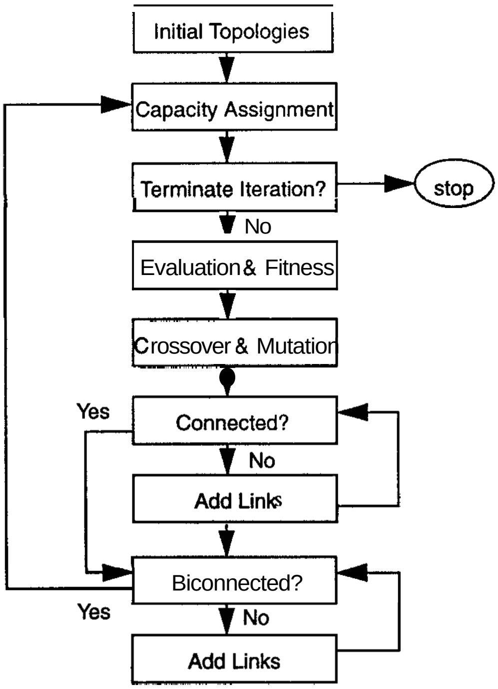

Fig. 2. The control flow of topology design with biconnectivity constraint

tive routes from i to j (since the topology is biconnected, the alternative route always exists). The default way is the first way.

In the following section, the results of applying this algorithm to a 20-node test case are presented.

2.4. Results

The algorithm has been applied to different network planning cases. The result **of** a 20-node network design problem is presented here. The traffic among the nodes is assumed to be uniformly distributed among all nodes. The link costs are obtained from [10]. The node cost is assumed to be proportional to the capacity **of** the node. The number of solutions in the solution pool is set to be 20. To test the performance of the algorithm under different settings, we alter the value of the crossover probability and the mutation probability. The costs of the best solutions at each iteration for different settings are shown in Fig. **3.** From the figure **we** ***can*** see that the probability of mutation has a large effect on the performance. **A** larger probability of mutation significantly alters the topology and results in a slower rate of convergence to the optimum. The probability of crossover also affects the rate of convergence. In this case, both the highest and the lowest crossover probabilities result in a slower rate of convergence. For all circumstances, as the costs of the best topologies at each iteration approach to the minimum solution, the convergence speeds slow down. It is also shown that a 15% reduction in the cost is reached after 3000 of iterations when crossover probability and the mutation probability are set to be 0.3 and 0.01 respectively. Other choices of the cross-

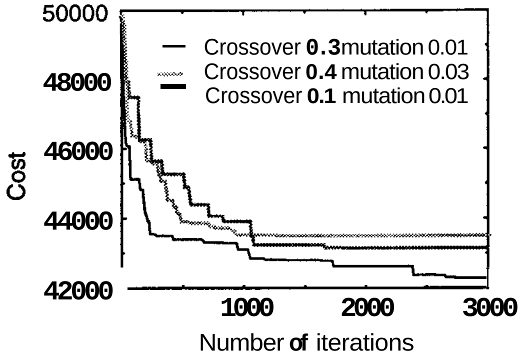

Fig. 3. Results of network with 20 nodes

over and the mutation probabilities perform slightly worse. Table 1 summarizes the results for the 20-node case.

This algorithm can also be applied to the planning case that must include the existing networks with minor changes. To force all the solutions to contain the existing network, we first require all the initial solutions (topologies) to contain the existing networks (subgraphs). Then, we change the mutation operator so that the mutation operations do not alter these subgraphs. Since all the solutions in the solution pool contain the subgraphs, the crossover operations will keep the subgraphs in the resulting topologies. Thus the main changes are the process of generating initial solutions and the mutation operator.

Table 1: Network Design Results

Iteration	Cost reduced	cost added for bi-connect.	Crossover prob.	Mutation prob.
3000	15.1%	0.6%	0.3	0.01
3000	13.1%	0.8%	0.4	0.03
3000	13.9%	0.8%	0.1	0.01

2.5. Conclusions and Future Work

We propose a layer framework for ATM network design. The layer approach makes the design task more manageable. Although topological design is still a large scale, non-linear, mixed-integer programming problem, we are able to solve the problem using a genetic algorithm. The algorithm is tested with a few test runs. From the test results, it is shown that the genetic algorithm we develop is effective.

The possible future work for ATM network planning includes the expansion of the algorithm to more complicated cases, for example, the case that the node locations are not given. In general, it is very difficult to solve this problem; however, for a special case that the possible candidates for the node locations are known, the genetic algorithm may still be applied, though some changes have to be made. The new requirement for this case is that the candidate nodes for one location are exclusive, i.e. only one of them should be selected into the solution. Thus, we can change the crossover and mutation operator such that for all node locations, only one **of** their candidates is selected into the resulting topologies.

2.6. References

[1] **R.** L. Sharma, “Network Topology Optimization”, ***New York: VanNostrand Reinhold, 1990.***

[2] Shau-Ming Lun, Felix Wu, Ning Xiao, and Pravin Varaiya, “NetPlan: An Integrated Network Planning Environment”, ***State of the Art in Performance Modeling and Simulation: Computer and Communication Networks, Vol.1, Kluwer Academic Publisher,*** 1993.

[3] **B.** Gavish, “A General Model for the Topological Design **of** Computer Network,” ***Proc. of GLOBECOM’86,*** pp. 1584-1588, 1986.

[4] A. Gersht and T. Weihmayer, “Joint Optimization of Data Network Design and Facility Selection,” ***ZEEE Journul on Selected Areas in Communications,*** vol. 8., No. 9, pp. 1667-1681, Dec. 1990.

[5] D. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” ***Mass. Addison-Wesley Pub. Co.,*** c1989.

[6] **Z.** Michalewicz, “Genetic algorithms + data structures = evolution programs,” ***Berlin; New York: Springer-Verlag,*** 1994

[7] **A.** Chuang, **“An** Extendible Framework for Genetic Algorithms in the Ptolemy Environment,” ***MS report,*** University of California at Berkeley, 1995.

[8] Ning **Xiao,** “Capacity planning and Topology Optimization of Corporate Communication Networks,” ***Ph.D thesis,*** University of California at Berkeley, 1993.

[9] T. **H.** Cormen, C. E. Leiserson, and **R.** L. Rivest, “Introduction to Algorithms:’ Cambridge, ***Mass.: MIT Press;*** New York Mcguire-Hill, 1989.

[10] Ricardo F. Garzia and Mario R. Garzia, “Network Modeling, Simulation, and Analysis,” ***New York: M. Dekker Inc.,*** 1990.

[11] D. Bertsekas and **R.** Gallager, “Data networks,” 2nd ed. ***Englewood Cliffs, N.J.: Prentice Hall,*** c1992.

3. Object-oriented environment

Abstract

This section explains why we abandoned the original aim of creating an integrated environment for modeling traffic flow, sensor data, and the communications infrastructure for ATMIS. It begins by summarizing the reasons for such an environment, describes the available software system support, and then explains why the original aim was impractical.

3.1. Introduction

Large engineering systems like the ATMIS system face the challenge of providing reliable services using scarce resources. Due to the heterogeneity of system elements (variety of transportation modes, roads and vehicles, user needs, sensors, decision entities, etc.), the planning and control of such systems can rarely be cast in a single mathematical framework, because such a framework requires a significant degree of homogeneity---that is, the existence of a unitary model in which all these system elements can be sufficiently faithfully represented. Most often one has a range of models for each system element and for combinations of elements that form mathematically representable subsystems. Thus, for example, one can have model(s) for vehicles, geographically accurate models of roadways, roadways represented by capacity numbers, microscopic and macroscopic traffic flow models, models for signals, models of communication systems, etc. These are models of some of the elements of an ATMIS system.

Each of these models is useful for certain limited purposes and in limited contexts. But it is impossible to combine these models into a single "unified" model in order to describe, analyze or design an **ATMIS** system "as a whole". The difficulties are both conceptual and technical. At the conceptual or semantic level the problem **is** that the variables defined in different models may not be directly comparable even when they denote the same things in the real world. For instance, in a microsimulation model like SmartPATH [13 individual vehicles are identified; in loop detector data one has vehicles counts but no means to identify individual vehicles as in the I-880 database **[2]** and in a traffic model one has densities (vehicles per unit length) and flows (vehicles per unit time) as in [3]. Although in a user's mind, there is a clear understanding which relates these three representations of vehicle, it is very difficult to "automate" such an understanding, especially since different users may understand these relationships differently.One technical difficulty is directly related to the different representations of the same real-world object, namely it is likely that different data structures are employed for these representations and there may be no unequivocal way of relating these data structures.

3.2. Object management systems

Object management systems (OMS) are object-oriented software systems used to simulate, evaluate, and control large-scale physical environments such as transportation networks and telecommunication networks. **OMS** provides these functions

- Configuration management--the ability to specify and control the configuration of the physical environment;
- Fault management--the ability to detect faults and other specified events in the physical environment and respond to these events;
- Performance management--the ability to track, optimize and fine-tune system performance;

- Resource management--the ability to provide an inventory of physical system resources; and
- Planning management--the ability to specify, simulate and evaluate alternative physical system configurations and policies.

The success of OMS depends on:

- The data and process models used to describe the environment;
- The software tools used to implement these models; and
- The software engineering process followed to implement the system.

The data and process models should capture the domain expertise required to describe and manage the physical environment. Typical modeling approaches use relational databases for modeling data and programming languages for process models. In complex applications, the object-oriented approach appears to offer superior modeling capability. While the relational model only describes system state **or** data, the object model can potentially be used to describe both system state and system behavior (or process), in an integrated manner [4]. However, this potential is rarely exploited in practice, and the object model is often used only for data description.

Because they have been widely adopted, the approach based on relational databases and programming languages provides a mature set of software tools. Relational databases offer a development platform including database engine, modeling tools such **as** form and report generators, and application development utilities. The relational model has a powerful Structured Query Language (SQL) whose semantics and mathematical basis are very well understood. It has been possible to develop a widely used standardized set of tools because of the structural simplicity of the relational model comprising a collection of flat, fixed-format tables. The popularity of the relational approach can be attributed to these tools

The role of tools in object-oriented modeling is even more important because it is semantically much richer than the relational model. It seems, unfortunately, that this richness has made it very difficult to develop tools of wide applicability. Object database developers have failed to agree on a common query language like SQL. The emerging object databases are of two types: those that are closely tied to the relational model of which Matisse and Postgres are examples, and those **tied** to programming languages such **as** Versant.

Since our aim was to build an environment in which we could integrate data sets (eg. 1-880 data), simulation models (eg. SmartPath), and complex system planning tools (eg. NetPlan [5]) we proposed to follow an object-oriented approach. We proposed an OMS model with the following features:

- State---An object's attributes describe its state, inputs and outputs. The system is an interconnected set of objects and the system state comprises the state of its objects and their input-output connections;
- Methods---The methods of an object model the object's dynamics, i.e. the ways in which its state changes;
- State transitions---These are discrete state transitions; and
- Constraints---These are constraints on state, connections, relationships between objects and behavior constraints.

3.3. SmartDb

SmartDb is a software implementation **of** the OMS object model. It provides the following additional features that we felt were essential to the communication and transportation network domains:

- It implements a relationship object and provides relationships such as input-output, containment, views, agent-manager, client-server and process layers. **(A** process layer is a collection of objects scheduled for execution at a common time or event.)
- It implements objects such as events, sensors, actuators, schedulers;
- It implements the OMS engine which executes the model dynamics; and
- It provides system architecture tools for data distribution, process distribution, object migration, process migration, etc.

SmartDb is built on top of a persistent storage medium, and requires these capabilities: persistent storage, schema generation, implicit retrieval, predicated queries, backup and restore, and commit and rollback. It is more useful with these additional capabilities: versioning, data distribution with location transparency, object migration, directory services, and concurrency control.

SmartDb was implemented using the C++ programming language, Versant Object Database, Tcl/Tk user interface tool kit, and the UNIX operating system.

As a test to determine how **good an** environment SmartDb could provide it was customized to form the SmartAHS platform for highway simulation. SmartDb and SmartAHS are described in detail in [6]. In a sense, SmartAHS was a more flexible (but incomplete) version of SmartPath. It was more flexible since the controllers that were "hard-wired" in SmartPath, were encapsulated **as** objects which could be changed without affecting the rest of the simulation environment.

The next question that we were faced with was whether to customize SmartDb to accommodate the network planning tool, NetPlan. For reasons explained next, it was decided not to pursue this direction of work.

3.4. Lessons

Two important, one positive and one negative, can be drawn from our **OMS,** SmartDb and SmartAHS efforts. The positive lesson was that an object-oriented approach provided a more flexible modeling environment for purposes of simulation than a "hard-wired" simulation package such **as SmartPath,** although the latter had a definite performance advantage. The features that permitted this flexibility are: object modeling, interconnections among objects, continuous and discrete state transitions. However, SmartDb was not a programming language but an approach, and consequently it did not provide the programming discipline that a well-disciplined language would enforce. Moreover, its semantics remained unclear since C++ was its specification language. This lesson led directly to the next generation of modeling framework, namely SHIFT, an object-oriented language for specifying a network of interacting dynamic objects with dynamically varying interactions [7]. **SHIFT** has been successfully used in many applications already.

The negative lesson was that an "integrated" environment can mean something as simple **as** providing a shell or a common interface to a diverse set of already developed tools (the interface providing a common "look and feel" as might be built using Tcl/Tk or a Netscape browser interface), or it may mean something **as** complex as providing a way to "interconnect" those tools, eg. by providing a means to combine simulation results with actual data, or data with performance evaluation tools, etc. Our aim was to aim towards the more complex notion of integration, and this we realized is very difficult, given the equivocal semantics of variables used in different software contexts even when they denote the "same" physical objects, insufficient specificity regarding how we wised the tools to interact.

One final lesson to be drawn is that commercial object-oriented databases do not yet match the support or performance offered by relational databases. Versant was chosen because it provides:

- Persistence through inheritance so that the application developer does not need to be aware of the persistent nature of the program;
- A library of persistent data types such as **lists** and arrays whose use speeds up the development process;
- Instance level blocking, essential for distributed simulation.

However, the following limitations hampered implementation of SmartAHS:

- Partial commits **are** unsupported;
- Partial **object** retrieval is not possible;
- There **are** no distributed directory services;
- Migration is unsupported in shared sessions;
- The Link construct cannot be subclassed to implement relationship semantics; and
- **The** server does not provide events or triggers.

3.5. References

[1] F. Eskafi, D. Khorramabadi and P. Varaiya, "SmartPath: an Automated Highway System simulator," PATH Technical Note 94-3, June 1994.

[2] A. Skabardonis, K. Petty, H. Noeimi, D. Rydzewski and P. Varaiya, "I-880 field experiment: Data-base and incident delay procedures," Transportation Research Record 1554,204-212, 1996.

[3] K. **Sanwal,** K. Petty, J. Walrand and Y. Fawaz, *"An* Extended Macroscopic Model for Traffic Flow," Transportation Research, Part B, 30B (1), 1-9, 1996.

[4] G. Booch, ***Object Oriented Design with Applications,*** Benjamin/Cummings, 1991

[5] Shau-Ming Lun, Felix Wu, Ning Xiao, and Pravin Varaiya, "NetPlan: ***An*** Integrated Network Planning Environment", ***State of the Art in Performance Modeling and Simulation: Computer** and **Communication Networks, Vol.1, Kluwer Academic Publisher;*** 1993.

[6] A. Gollu, "Object Management Systems," PATH Research Report UCB-ITS-PRR-95-19, June 1995.

[7] ''SHIFT The Hybrid System Simulation Language," available at: **`http://www.path.berkeley.edu/shift`**

4. Conclusions

The original aims of this MOU were to create an integrated environment for modeling traffic flow, sensor data, and the communications infrastructure for ATMIS. The plan was to build separate tools for measuring the communications requirements of ATMIS traffic, modeling a mobile radio communications environment, and communications network design, and then to integrate them in an object-oriented environment from which one could also access existing or new road traffic simulation packages.

We developed a framework for evaluating the communication requirements for **ATMIS.** The framework fits within the National **ITS** Architecture and the California's ITS program. We have begun a case study for the San Francisco East Bay Area. That study should be completed by September **1997.** We extended the NetPlan environment for designing communication networks by providing tools for the topological design and capacity planning of **ATM** networks. We provided a framwork for wireless radio networks, focusing on channel models. We customized the object-oriented system SmartDb to SmartAHS---a microsimulation package for automated traffic. We succeeded only partially in building an integrated environment within which different tools could be inter-operated. From this partial success we learned important lessons that led to the successful development of **SHIFT.**

Based on the work done under Mou **141** one may draw some more general conclusions. First, planning the deployment of **a** complex communication system for ATMIS is best conducted within the framework provided by the National **ITS** architecture, keeping in mind the lack of adequate forecasts, the underdevelopment of markets for **ITS** services, making use as far as possible existing communications technologies. Second, it is very difficult to integrate simulation tools that have been independently developed and possess incompatible variables and definitions. Third, the object-oriented approach to simulation provides a great deal of flexibility, but care must be taken to enforce programming discipline and to provide strict semantics, even at the cost of a loss of flexibility.