OPNET: An Integrated Design Paradigm for Simulations

Sparsh Mittal

Electrical and Computer Engineering, Iowa State University, Ames, IA, USA 50011

sparsh@iastate.edu

Abstract- **In recent years, a lot of progress has been made in the field of networks and communications; and also in design of simulators. In this paper, we survey and review prominent fields where OPNET has been applied and compare it with other existing simulators. Our work helps beginners and researchers alike in estimating the useful features and limitations of OPNET and the state-of-art in network simulations, modeling and curriculum design. We cover a wide variety of areas of application, to highlight the versatility of applications of simulation tool. The conclusions from the study give valuable directions for further employment of OPNET and also for its extension.**

Keywords- **OPNET, Simulation, Networks, Curriculum Design.**

1. INTRODUCTION

Simulation modeling is becoming an increasingly popular approach to study the functionality and performance of proposed models in several fields [1–4]. Their usefulness and power in providing the ability to quickly experiment with multiple design options in the design space, user-friendly interface and enabling simulation of the dynamic nature of the network make them preferable over mathematical analytical modeling. Several simulators have been designed for use by the academic community students and researchers today. Some of them include REAL, NetSim, Harvard Simulator etc. In this survey we choose OPNET (OPtimized Network Engineering Tool) [1] and present a survey of its use and features. We also summarize the relevant features which are of particular interest to different user-groups.

Since it is impossible to do justice to this large body of research in this short survey, in what follows, we discuss some of the studies most relevant to our proposal. Some of the features of our survey are:

1) We survey the research progress from a wide spectrum and level of work, such as technical reports, experiences of teachers with teaching OPNET, OPNET in state-of-art research etc. This highlights various functionalities of OPNET.
2) This paper underscores power of OPNET as simulation tool and the crucial role network simulators play in research. We believe that this survey will help designers, architects and teachers to get an overview of OPNET in different domains of applications.
3) We focus on exploring the salient feature and limitations of OPNET, which become especially clear when OPNET is employed in a research project.

2. OPNET IN CURRICULUM AT UNDERGRADUATE AND GRADUATE LEVEL

In several fields of research, user (who may be a customer or student) satisfaction is being seen as an important criteria for measuring performance and several studies have been conducted to develop evaluation models and measure the user-satisfaction; and provide this information as feedback to take decisions about system parameters. Similarly, the use of OPNET is courses have also been widely evaluated. The experts have also strongly favored laboratory components as essential components of a networking curriculum and stated that a thorough understanding of networking requires laboratory facilities to enable the students to build, observe, experiment and measure [8].

The authors in [5] discuss their experiences with laboratory and classroom projects using OPNET and Wireshark in undergraduate networking courses, such as Data Communications and Networking, TCP/IP and Internet Technologies etc. They have found the tool as a great help in enhancing student learning and bringing student interest in computer network courses. In such courses, typically OPNET is introduced along with a classroom demonstration and step-by-step explanation through tutorials for systematic learning of each concept, such as basic steps and software specific features, modeling, running the simulation, debugging etc. They have found that the students benefit greatly from an immediate homework assignment to create and simulate their own simulation in OPNET and moving to next lecture after reviewing the previous exercise. The authors also present a brief listing of the few OPNET projects offered in their undergraduate courses, beginning from basic introduction of OPNET features up to the level of using OPNET to simulate complex network and independent design and modeling based on a high-level description of a networking system. The feedback from the student allowed the instructors to assess the specific problems faced by the students such as monotony of following step-by-step OPNET configuration instructions, difficulty of debugging and complexity of result analysis and to re-write the lab manuals and restructure the labs based on it.

The authors in [6] present their approach to develop an OPNET simulation networking laboratory to reinforce classroom lectures. Through OPNET, lessons on the dynamics of network protocols, and various networking aspects such as the design and the limitations of protocols, simulation and performance evaluation techniques, interpretation of data and packet analysis etc. are imparted to the students. The authors have found that the open design of the labs encourages active learning and such an active learning approach gives the students a valuable experience. This is also likely to help them in understanding the subtleties of the design of a complex system for future use. This is in accordance with the IEEE/ACM Computing Curriculum 2001 [7] which recommends integrating hands-on experimentation and analysis into networking courses. The authors discuss the specific design objectives of simulation lab and discuss the desired properties of any network simulator for being used in such laboratory experiments. The courses offered at introductory levels are of less-specific nature. Thus, they cover a relatively wide variety of topics in networking, instead of focusing on a few topics which may be the case in research labs or industries. Hence, as a simulator, OPNET is found to be useful because of its ability to simulate a wide spectrum of networking technologies. Also OPNET enables modeling the entire network, including its routers, switches, protocols, servers, and the individual applications they support.

For teaching at undergraduate and graduate level and for research purposes, a lot of universities are now using OPNET. Leeds Metropolitan University in UK is one such example [10]. Several graduate level courses use OPNET, such as Network Systems Infrastructure Unit, Networked Multimedia Communications Unit, and Mobile Communications Systems Unit etc. In such courses, OPNET is used to help the students critically analyze, evaluate and explain the behavior and operation of contemporary data communication systems at an advanced level. Moreover, students also explore the facilities provided by the simulation tool to study the interoperability and performance issues relating to homogeneous and heterogeneous data communications environments and to design, simulate and evaluate appropriate networking solutions to a range of real life data communication problems. This gives the students valuable hands-on experience and lets them think in terms of practical application. At undergraduate level, OPNET is used for teaching communication courses.

The authors in [22] share their experiences with teaching computer networks at West Point, New York through modeling in OPNET. For the undergraduate level courses, OPNET nicely does the task of modeling through visualization. As a communications oriented simulation language, OPNET has the feature of providing direct access to the source code and an easy-to-use front end. OPNET models are composed of three primary model layers, namely, the process layer, the node layer and the network layer. The ability to build process models using finite state machines encourages a modular approach, since the complex networks can be broken down into individual states and further, each state can be individually defined and implemented. At the next level of abstraction node models are used which could be a predefined OPNET artifact or defined by user. Above this in the hierarchy lies the network model, which may represent a hierarchy of sub-networks. Such an approach is especially useful for the students for gradual learning.

The network modeling in OPNET properly serves the pedagogical purposes of supervised laboratory projects and individual homework assignments. The functionality to access the source code and design new models encourages creativity and learning in the students. The authors have found that in-class animations, homework exercises and laboratory projects greatly help the students to visualize and therefore quickly understand the effects of parametric changes to a network design. This low-cost environment for active learning fosters a better understanding of the network through modeling and experimentation. Additionally it gives guidelines to choose the best among the various possible design options.

3. OPNET FOR MODELING REAL-LIFE SITUATIONS

Today, the applications of communications and networking systems are wide-spread, and the areas include military, health, education etc. [30]. The Multidisciplinary University Research Initiative (MURI) project [11] undertaken at the University of Michigan is aimed to investigate the various algorithms and protocols at different layers of upcoming army communication systems. As a powerful and flexible tool and the choice of industry, OPNET is used here as the simulation tool. The project has an OPNET simulator and an optimization module which communicate with each other. For each parameter set generated by the optimization program, OPNET simulator runs the network model of MURI many times with different random seed and returns the RMS error as the performance measures. This process is performed iteratively till a given number of iterations in the optimization program are reached. To cope up with the computation intensive nature of the simulation, OPNET simulations are run in parallel over different machines with different configurations. This greatly reduces the simulation time and allows the researchers to build more complicated networks.

The authors in [12] discuss an example of use of OPNET for simulation and performance analysis of the campus network of SUNY College at Fredonia, New York. To meet the increasing demands of the network resources, the college has adopted switched Ethernet subnets and Gigabit Ethernet backbone. However, to further test the ability of the network to support VoIP or videoconferencing applications, research is carried out to evaluate the performance of the network. Their work deals with performance measurement and improvement in network capability using innovative algorithms and identification of potential bottlenecks in communication. The authors also model the campus network using OPNET. For their work, the various features provided by OPNET come very handy, such as custom model creation to model the elements not present in standard node/link models, availability of a large number of models for link, nodes to choose from for routers etc. They then modeled the various details of clients, routers and networks lines on the campus, making suitable assumptions. To test and verify a number of experiments were conducted such as simulation for voice traffic between two clients, overall campus network simulation with existing and estimated future communication requirements. The results from the experiments were positive, matched the expectations of network managers and reflected good network performance. The results on traffic and congestion gave clear directions for the steps to be taken e.g. the nodes (i.e. campus locations) where the link capacities have to be increased in the future etc.

4. OPNET IN RESEARCH: FOR DEVELOPMENT, TESTING AND VALIDATION

For addressing the high computation demands of recent state-of-art application domains, researchers are employing efficient hardware platforms [33], versatile techniques [34] and simulators etc. Thus, due to its efficiency and detailed modeling, OPNET can prove to be a useful tool for researchers.

The authors in [4] provide implementation details of modeling in OPNET along with examples of some sample networks. They highlight the key features of OPNET and provide a summary of a few data editors and design flow in OPNET. OPNET provides four tools called editors to develop a representation of a system being modeled. These editors, the Network, Node, Process and Parameter Editors, are organized in a hierarchical fashion, which supports the concept of model level reuse. Models developed at one layer can be used by another model at a higher layer.

OPNET enables simulation of a large variety of networks, ranging from a small network to a large network consisting of many components, protocols, routing strategies etc. The authors in [13] present an OPNET simulation of random access CDMA system. CDMA based MAC protocols have the ability to reduce collisions, typical in random access networks such as ALOHA. The authors simulate un-slotted random access CDMA systems, since compared to the slotted version, these systems present much more challenge in their performance analysis because of the changing number of interfering users during the packet interval. The authors developed a MATLAB CDMA simulator to generate BER tables, however this approach was found to be very computationally intensive, hence a Simplified Improved Gaussian Approximation (SIGA) is used for its accuracy and simplicity. The model has two parts, namely pipeline stage and receiver-transmitter nodes. The MATLAB simulations were performed in tandem with OPNET simulations. MATLAB is used to generate the bit error rate tables and then, these tables are included into the OPNET simulation. (BER vs. number of interfering users). The results are also calculated using the SIGA method in OPNET. A comparison of the two shows negligible error, moreover, these results also agree with the theoretical results. It is found that by using CDMA, compared to an un-slotted ALOHA packet network, the maximum throughput is increased takes place at a higher offered load.

The author in [23] discusses the work on ATM Network Simulation with OPNET. Asynchronous Transfer Mode (ATM) is a connection-oriented packet switching technique that is universally accepted as the transfer mode of choice for Broadband Integrated Services Digital Network.

The ATM model suite in OPNET contains several client and server node models, such as workstations and servers, uni-clients and uni-servers, uni-sources and uni-destinations, and intermediate switching elements (such as clouds and switches). To specify the configuration details of ATM network, some parameters such as traffic contract, quality of service (QoS), port buffer configuration, available bandwidth, switching speed and routing attributes etc. can be specified. The author conducts experiments to study various parameters of the simulated network, using ATM distance vector routing protocol, and different queuing schemes, such as round-robin and weighted round-robin. The results show that network performs almost similarly with these two queuing schemes. The study also furnishes the values of several important parameters of ATM circuit.

The authors in [16] employ DFT (Discrete Fourier Transform) and other frequency analysis techniques to the design of intrusion detection algorithms. The frequency-based detection strategy is demonstrated in OPNET by running synthetic network intrusion data in simulated networks. Among the many approaches proposed in the literature for the design of security systems for monitoring the network and identifying the intrusion, frequency based technique uses the characteristic that network attackers generally use brute-force approach of running pre-written scripts to automate the process of making fake connections or forged packets. Thus by finding the periodicity or other statistical pattern (such as inter-arrival time of packets, size of packet payloads, or other anomalous traffic behavior etc.) within the traffic data, an attempted intrusion can be detected. For different simulations using different data sources for network traffic, suitable pre-processing tools are used to extract the traffic information that is necessary for OPNET simulation. Such information include data packet headers, flag information etc. In OPNET, a firewall node is used between attack victim node and the hub. The firewall node is used for monitoring the suspicious data traffic and testing the intrusion detection strategies. The attacks are simulated and simulation results for the network are collected. The results show that an attack is accompanied by a sharp increase in the number of port connections to the victim node. Moreover, a sudden increase in the number of data packets hints the presence of data packets that are suspicious of attacks. The frequency plots of the connections show distinct peaks corresponding to the attack. Based on these studies, even the IP address of the attacker can be identified. The authors are also able to detect attacks such as DOS attacks, password guessing attacks, Dictionary attacks etc.

The authors in [17] use OPNET for simulating Fiber Distributed Data Interface (FDDI) and Asynchronous Transfer Mode (ATM) technologies based network and for developing process model for leaky bucket congestion control algorithm in packet data network.

FDDI is a networking technology that supports 100 Mbps transmission rate, for up to 500 communicating stations configured in a ring or a hub topology. The authors have simulated the network scenarios with single attachment stations connected in a hub topology with FDDI concentrators in OPNET. The FDDI model is available in OPNET model library and allows the users to select various parameters. For different model attributes and parameters, the authors measure the end-to-end delay variation with the load and throughput for the network. It is found that the end-to-end delay in the network decreases with increasing network load and finally levels off, showing stability in the network. Thus, along with the appropriate information about value of network parameters, OPNET models can also give useful hint about stability etc. for the network.

For the ATM network, the requirement of QoS for different network varies depending upon the type of applications running on client. For example, a voice application (telephone conversation) may demand transfer delay being small, however for video applications, delay jitter is more important a parameter than the transfer delay. For a data transfer application, accuracy is of high priority, which may not be so significant in a video application. Thus to match the needs of different applications, different service categories are supported, such as constant bit rate, available bit rate (ABR), real time variable bit rate etc., each having its own advantages and disadvantages. Among these OPNET model library supports an ATM client-server network with CBR and ABR clients and other service categories are also simulated. Based on the response time metric, the CBR is found to be the best among all.

In ATM networks, the bandwidths in the channels are not fixed, thus the congestion scenarios can occur because of excessive use of negotiated bandwidth. This may lead to several problems such as high delay, cell loss, degradation of network functionality etc. For addressing this problem, a policing mechanism called leaky bucket [a18] is used for connections with CBR traffic. The algorithm for leaky bucket is implemented by using process model in OPNET, with five states. A process model has a state transition diagram which is somewhat like a finite state machine. OPNET provides advanced GUI based editors for creation of process model. Starting from initial state, the process can either reach arrival state when a packet arrives, or idle state where it remains until the next cell arrives. From arrival state, the process reaches either serve or drop state, depending on whether the bucket is full or empty, respectively.

For the networks with CBR traffic, the leaky bucket only monitors MCR (mean cell rate). Analogously, a dual leaky bucket mechanism is used in VBR traffic, where they monitor PCR and MCR of the source respectively. The leaky sources are used to police the sources which have negotiated some MCR and PCR but are likely to not confirm to it. Simulations are performed to collect the number of packets that are discarded by the leaky bucket and the number of free spaces in the bucket at different instances is calculated. It is found that the size of the burst entering the network is limited by the bucket size and also, the number of lost cells is a function of the leaking rates. The results match the expected results from theoretical calculations.

The authors in [19] develop simulation model of Mobile IP using OPNET. Mobile Internet Protocol (Mobile IP) [20] is employed to provide seamless support for routing of IP datagrams to mobile hosts. An efficient and satisfactory implementation of MIP involves use of several mechanisms such as movement detection, buffering, bicasting for handover smoothing etc. However, there are several issues which need to be addressed such as handover performance, security, and integration of QoS etc. Such requirements are necessary for enabling real-time multimedia traffic, preventing security attacks in a highly vulnerable wireless environment and also to provide good mobility and support to the users.

The development of the complete network benefits greatly from the modular design strategy supported by OPNET. With such an approach, separate "Node Models" are developed for different entities that would be used in the network, such as mobile node (MN), host agent (HA) etc. Each node model consists of several component modules such as processor/queue modules and streams providing connections between modules. Every node should have more than one physical link, e.g. Ethernet, Point-to-Point, or WLAN (Wireless Local Area Network). The most important component module of a node model is the processor module which implements the role of the given node in the network, which is coded into a 'Process Model'. The examples of such process model are the MN process model and the HA process model that are used in the MN, and HA node models respectively. The authors also identify the functionalities of MIP, which are supported in OPNET and those, which are not supported.

For supporting Mobile IP, the authors implemented several mechanisms such as agent discovery and movement detection (ADMD), home registration and deregistration, and IP tunneling (Encapsulation and decapsulation) mechanism. Wireless LAN is configured as IEEE 802.11, with 11 Mbps data rate. The WLAN radio coverage is set to 250 meters; that ensures non-overlapping radio coverage of separate APs. Mobility pattern of the MN is characterized by a horizontal linear path with a fixed ground speed of 30 km/h, and for observing the impact of moving speed; it can be varied from 1 to 30 km/h. The application traffic exchanged between the CN and the MN is configured to represent IP Telephony using Voice-over- IP techniques. The voice traffic is bidirectional and random in nature.

The results from the simulations provide valuable conclusions. Among the various ADMD methods, for moderate path delay of value less than 150 milliseconds, ECS (Eager cell switching) or ACS (Active cell switching) is strongly preferred, since they lead to small value of average handover latency. The experiments performed with variable moving speed of MN, show that the with an increase in the moving speed of MN, the overall throughput goes down. This is expected and is attributed to a corresponding increase in the signaling load and to the increase in disconnection time over full communication time due to handover. On higher speeds, the handover rate increases and this also leads to a larger signaling overhead per unit time period. It is also observed that the buffering mechanism used ensures timely delivery of user traffic as soon as MN begins registration through newly discovered FA. As the FA-HA path delay increases, the throughput decreases since the higher path delay restricts the amount of user traffic along the HA-FA path from CN to MN.

Buffering ensures better performance since it alleviates the impact of packet loss; however it may increase instantaneous packet delay variation which is undesirable with real-time applications, such as Voice-over-IP. In absence of buffering, regional registration improves performance since it reduces handover latency and packet loss. On the downside, it may incur up to 33% more average signaling load. Such disadvantages may sometimes outweigh the benefits of regional registration when bandwidth constrained wireless links are involved.

The wireless networks are examples of multi-stage interconnection network and their performance largely depends on the factors such as topology, traffic distribution etc. Among the various switching methods used, flit-based switching [21] offers the advantage of simplified implementation and simpler signaling constraints and lower latency. The authors in [20] discuss an implementation of flit-based network. They build a low latency network using spatial frequency division multiple access (SFDMA) protocols for reducing the need to contend for a channel; moreover, flit-based networks are used to reduce latency. OFDM technique is used for enabling the use of multiple frequencies and capability of simultaneous transmission and reception [20].

The simulation environment is developed in the OPNET. Wireless communication in OPNET is realized using the radio transmitter and receiver modules provided by OPNET. These modules allow variation of the various parameters such as the bandwidth and minimum frequency, the data rate, and the modulation of the channel. The authors developed a FLIT_RELAY model with 48 channels configured for the transmitter and receiver modules, for modeling the subchannels of OFDM. It is required that for each of these channels, the bandwidth and minimum frequencies of the transmitter and receiver match. The modulation used is QAM-64 and the data rate is set to be 1.125 MB/s for all channels. The simulation results suggest that using the relay technique, a very small value of packet latency can be achieved than what is achieved using the conventional 802.11 communication. In fact, the introduction of the flits increases the gains even further, since the smaller sized flits can be pipelined through a multi-hop network and such pipelining further reduces the packet latency in comparison to a simple store and forward approach.

During the implementation of OFDM channels, the authors face a bottleneck of the OPNET modeler. In OFDM packets can be modulated over a number of subchannels that are not adjacent in frequency space. However, the OPNET channel model does not allow this, as the interference model would be compromised. In OPNET, a packet can be sent to a single output channel. This is a fundamental modeling concept in OPNET. To strike a solution, the authors used 48 independent subchannels. To ensure highest channel utilization using independent subchannels, very small flits are transmitted which more evenly spreads the packet across the subchannels; however, use of smaller sized flits also leads to increase in the simulation time and increased overhead due to the labels. The experiments performed to study the effect of varying flit size shows that with subchannel aggregation, change in flit sizes makes little impact, however, with independent subchannels, variation in flit sizes leads to wide variation. Overall, the research conclusions suggest that employing subchannel aggregation along with smallest flit size that simulation time allows, gives the best results.

The authors in [25] discuss the design and implementation of differentiated service routers in OPNET and study UDP performance over Diffserv in a large scale network. Recently developed applications such as distributed multimedia and real-time applications increase the demands for Quality of Service (QoS) guarantees. This presents a challenge for internet since internet only provides one simple service class with respect to QoS, which is best-effort datagram delivery without any service quality guarantee. The Integrated Service Model (aka IntServ) [26] provides an integrated infrastructure for handling conventional Internet applications and QoS-sensitive applications together. Although, along with RSVP as the signaling protocol, IntServ can provide QoS guarantees to the applications, however, it does not scale well. To address such issues a stateless model, called Diiferentiated Service Model (DiffServ) [27] has recently become popular and widely studied. Before its actual deployment at wide scale into the whole internet at one shot, simulation is the best way to study and experiment with DiffServ.

The primary purpose of the Diffserv model lies in provisioning end-to-end QoS guarantees by using service differentiations in the Internet. The Differentiated Service enabled routers (DS-enabled routers) are key nodes in the Diffserv model, which are of two kinds, namely (1) edge routers and (2) core routers. In their experiments, the authors use edge routers. For experimentation purposes the authors constructed a large-scale network environment with multiple DS routers and traffic senders/receivers for verifying the performance of DS routers. The model simulated has multiple DS routers, traffic senders and one receiver. The model has client and server subnets with its own routers. Compared from the uncertain and error-prone development style of starting everything from scratch, the OPNET libraries provide the option of choosing the existing nodes. Moreover, since the nodes have a large number of functionalities implemented already, programmers save a lot of time. However, OPNET is not rigid and gives the user sufficient flexibility and power to extend its functionality by re adding user-written code, wherever appropriate. The DS scheme is put into the IP module.

Due to the object oriented approach of OPNET with well-defined interfaces, the changes made in one module minimally affect the other nodes or their functionality. Thus, merely by keeping an appropriate interface, the whole model works fine, as before. The process model for DS enabled IP module in OPNET has two different processes, namely, the main IP process which implements main IP and Diffserv functionality, and secondly the child process which implements priority scheduling scheme for Diffserv. To implement it fully, various issues such as classification of packets, implementation of packet monitoring and policing schemes, packet routing and forwarding schemes are also addressed. The leaky bucket model and token bucket model are used for conformance checking on premium class traffic and assured class traffic respectively.

For experimentation, the "Video Conferencing Transport" provided in OPNET using UDP as the transport protocol application is used. Its scenario consists of, first, the client (traffic sender) sending UDP traffic to the server at a constant rate and second, the server (traffic receiver) echoing the traffic back to the sender at the same rate. Any simulator, however detailed will always have some limitations, and OPNET is no exception. The simultaneous use of both DS-enabled and non-DS-enabled IP modules is unsupported in OPNET and thus generates compilation error message. To address this, the authors accommodate both IP modules in same module and provide an additional flag to the user to differentiate between these two modules, whenever necessary (such as upon packet arrival).

Another problem faced was due to the thrashing phenomenon, because of large size packet getting fragmented into multiple IP packets in the IP layer during transporting and sending rate exceeding the speed limit. This leads to video applications not getting completed since even if one defragmented packet is dropped; the whole application packet is discarded. The reason for the error was keeping the monitoring and reporting modules in the application layer of the traffic receiver. To address the problem, these modules were moved to IP_encap layer. This solved the problem and also led to obtaining more accurate IP statistics for different service classes.

The simulations verify the correctness of the PS-queue and RIO-queue implementation. Also, the experiments performed for testing DS-router gives positive results, confirming to the expectations and complying with the DS principles. The work by the authors can be extended to evaluate TCP performance (which is more complicated than UDP, especially in a Diffserv environment), and building up a multi-domain DS simulation environment. This will allow further research on Diffserv topics such as pricing scheme and routing protocols etc.

5. EXTENSIONS IN AND OPTIMIZATION WITH OPNET

The authors in [24] use an API along with OPNET model based event controller as the basis in creating an Air Traffic Control scenario from detailed behavioral models. An important advantage of distributed simulations is that, compared to stand-alone scenario, it provides opportunities for speed-up.

In the stand-alone mode, all the aspects of simulated entities are computed and managed within the OPNET modeler system. This approach suffers from the artificial need for OPNET to capture the behaviors of many non-network related components such as mobile nodes etc. To achieve precision in such modeling is a computation intensive and costly task. This task could better be distributed to other specialized and already-existing simulators. Thus to optimize such simulations, the authors argue for a distributed approach where OPNET is employed to exclusively model scenario communications behavior while external simulations collaboratively manage non-networking issues. Such an approach minimizes the workload of resolving communication effects and improves the quality of overall results. This however requires careful management of interdependent simulations to maintain cooperative operation. This includes taking care of issues such as data format, protocols and time management to maintain synchronization and causal relationships. Some of the other issues include simulation object management to guarantee compatibility and equivalence among objects of distributed simulation.

Considering the practical issues such as budget and time available for development, the authors decided to create a simple and flexible distributed interface for OPNET to allow an inexpensive exploration into distributed simulations and dynamic network loading. Their distributed architecture is primarily composed of two parts, namely, the Comm API and the Controller. The Comm API library provides a set of distributed simulation services to all collaborating simulations such as time management, data exchange and entity updates etc. The Controller, as an OPNET node model, interfaces an external simulation with OPNET and also communicates to the entities within OPNET, through 'controller_if' nodes.

The controller coordinates the simulation distributed execution. The controller can manage any types of assets and message types which are specified by the Comm API. The Controller model can be added to a network simulation using the standard OPNET drag-and-drop style and can be dynamically switched on or off. The Controller updates position and velocity attributes of each element in the simulation through remote interrupts to each of the players. Through the Comm API, the Controller receives network loading requirements from the external models and co-ordinates among the network objects. Finally, the Controller relays communication statistics collected in OPNET to external models, and is used to determine future player action in their portion of the distributed simulation. The control exchange among the Comm API and OPNET kernel ensures a proper event driven execution through interrupt mechanism. Thus the Controller ensures that the Comm API and external simulations work in concert. The Comm API allows distributed simulation between OPNET and an external simulation. The flow of information through the Comm API is controlled through a standardized series of parameters measuring time and communication events.

The authors also discuss the extensions required to reach the HLA (High level architecture) collaborative capabilities. As an example for the proof of concept, the authors created a scenario for a distributed commercial aircraft controller; such a scenario has several simple behavioral models defined in an external scenario, such as air traffic control tower, control tower radar, commercial aircraft, ground based radar etc. Here, based on the success or failure of the communication, an acknowledgement is sent or appropriate action is taken. The results from the experiments matched the expectations and the distributed approach successfully provided a way to for dynamic loading of OPNET scenarios and real-time network loading QoS metrics to external simulation tools. The slow down due to the Comm API and the controller was small enough to be acceptable.

Further, a hybrid architecture using both statistical traffic sources for predictable background traffic and reactive traffic sources for traffic explicitly initiated by the behavioral model was also tested. A common observation from these experiments has been that with the increase in the number of nodes, the number of position updates between OPNET and the behavioral simulation increases proportionately. Overall, the distributed approach demonstrated the benefits of incorporating OPENT communications link performance models within a larger distributed simulation.

6. OPNET: COMPARISON WITH OTHER SIMULATORS

To study and explore the relative advantages of different simulators, several studies have been conducted. The authors in [14] compare OPNET modeler with another popular simulator called Ns-2 from the Virtual Internetwork Testbed project VINT [15]. The results from the real world network simulations are compared with the simulation results from these two simulators and it provides valuable guidelines to network researchers.

The comparison among the simulators demonstrates the framework for validating the suitability of simulators for the specific case of packet level forwarding in an IP transport network. The network simulators are primarily divided on the basis of methods of simulation and fall into two categories: discrete event or analytical simulation. The general commercial simulators such as OPNET and Ns-2 are called hybrid simulators, since they combine both methodologies to provide a reasonable speed and, at the same time maintaining accuracy in the critical areas. Ns-2 is an event driven network simulator and used primarily by networking research community.

The network testbed used for the experiments consists of five computers, two of them operating as client server pair, another two as traffic-generator/traffic-sink pair and the fifth one as a router. The authors compared two scenarios, one for CBR experiment and another for FTP session. The CBR traffic is characterized by a fixed bandwidth across the network and is typically used by applications such as video and audio while as FTP is intended to share, transfer and transmit information between two computers. For the two simulators, different design parameters were varied to test their impact on the performance. The simulator was tuned for accuracy and their outputs were compared to the output from a live network testbed.

One downside of the network simulators was that due to inherent limitation in the degree of closeness possible in modeling, the simulators are always somewhat different from the real testbeds. The actual systems have a great number of variables that control actions such as packet forwarding and exact modeling of all these is practically not possible. The simulation speed of both the simulators is high and thus they are quite fast. Compared to Ns-2, OPNET provides more diverse statistics modules at different levels, although this also adds to the overhead of the software. Ns-2, being a freeware is likely to be a choice of many students while as OPNET, due to the requirement of a license, would not be the as attractive. However, OPNET has a well-engineered GUI suitable for a commercial product, while as Ns-2 models require acquaintance with the *tcl* scripting language. The authors also found that for the CBR data traffic, the simulators had no significant problem in modeling the testbed; however the simulators did not adequately model the dynamic behavior of FTP, since the response of the simulators did not closely mimic the performance of changing network conditions. Overall, the comparisons such as this provide many useful insights.

The authors in [28-29] test Ad-hoc network protocols and suggest optimizations using OPNET. The Ad-hoc networks are today increasingly preferred over GPS (Global Positioning System) because of their many advantages. To fulfill the navigation and positioning requirements with Ad-Hoc network, the authors optimize its sensor network configuration using OPNET. The optimization is done in terms of making it fast, use less number of hops per route, low power demand and low error rate.

In an Ad-hoc network, the nodes are able to form a network automatically. The routing requirement of the source node is passed to the destination node. Now, the requirements for navigation and positioning in the network are many. Firstly, the navigation is distributed in a very large network of low memory and low bandwidth nodes. Partitioned areas make centralization impossible. To reduce the power demand, the amount of communication and computation requirements have to be reduced. Also, the positioning system should have an optimum number of sensors, and the protocol should adjust the topology and sensor settings to the changing environment. The authors show the required bound on the values of parameters of the network, to determine the best number of sensors in a specified area and the number of sensors for an acceptable delay and also the maximum number of sensors. The delay is varied with the number of sensors used and the simulation results match the theoretical calculations regarding the number of sensors required to be used. Thus simulations using OPNET provide accurate guidelines for implementing an optimal network topology for future navigation systems where GPS may not be acceptable.

7. ANALYSIS AND INSIGHTS

In this section, we present the insights gained and the specific features of OPNET which benefit specific user-groups. We believe that this analysis will help policy makers and curriculum designers to evaluate benefits of OPNET and pinpoint the specific features useful for different user-groups.

A. OPNET in lab curriculum

We found that the most important requirements of any simulation tool used in undergraduate level labs are small learning curve, graphical user interface which stimulates the interest of beginners, small development time of student-level projects and ease-of-programming (not error-prone). Since the students have to initially carry out prescribed experiments/simulations, the richness of library is less important concern; moreover, since the use of the simulator is for academic purpose, and only one time purchase is made; the cost of license purchasing is also less serious a concern. Requiring the students to use one tool in school and a completely different one in future (in industry or research) would be confusing and a gross waste of time. Instead to allow the student to continuously build upon the knowledge, they should preferably be taught a tool which they can later use in their professional life. Some features of OPNET appealing for this user-group are:

1. The GUI makes the modeler very attractive and easy to learn.
2. The OPNET university program has helped in wide-scale acceptance of OPNET in leading universities.
3. Opnet is very popular within academia, commercial and industrial communities. Thus a student in the field of networks need not learn many tools.

B. OPNET for researchers and independent-developers

A researcher would be most interested in a simulator, which is accepted by the research community and where the experiments can be compared with the designs/ideas already existing in the literature. Thus, in research communities, the general trend is to use standard simulators. A researcher would be interested in testing his/her idea for a variety of nearly-similar (but different from each other) design options and meaningfully compare them, to choose the optimal design. A researcher would be proposing novel ideas and would most often use rare (not common) features or components from the simulator; thus the preferred simulator should have a rich library with many existing designs. Moreover, the independent developers would also be interested in extending the functionality of simulator and would need systematic and detailed documentation. Thus compared to a tool, with no possibility of its extension (e.g. due to unavailability of source code); they would prefer a tool which gives them flexibility for its extension or modification.

We found these features of OPNET especially suitable for such users:

1. OPNET provides a development environment for the specification, simulation and analysis of networks from different fields and with different levels of complexity.
2. OPNET IT Guru is a network simulation package which allows a user to use point-and-click to design simple or sophisticated network systems, and analytically study performance of the network.
3. OPNET Modeler additionally allows the user to modify existing system components and create new ones. The engine of the OPNET Modeler is a finite state machine model in combination with an analytical model.
4. OPNET can model protocols, devices and behaviors with about 400 special-purpose modeling functions.
5. The detailed documentation and a large number of study cases
6. The functionality to create multiple scenarios to create different design variations helps to create closely related designs.
7. The functionality and the models available are regularly increasing with the use of forum named, "Contributed Model Depot", where different users/researchers can submit their models for others to use.

8. The OPNET model in its very core consists of C++ codes. These codes are compiled and executed just like the C++ program. This enables very detailed control of the model by the user. Moreover, C++ is a popular language.
9. Numerous studies have been conducted, using OPNET for building and testing models of academic and research interest. Such studies give the researchers significant directions and overview of the state-of-art.

C. OPNET in industrial communities

In commercial applications and mission-critical applications such as military etc., reliability of the simulator for testing and development is required, since the loss due to the use of an inaccurate or faulty tool is quite heavy. We found that these features of OPNET would be more meaningful in such applications:

1. Instead of an uncertain and error-prone development style of starting everything from scratch, the OPNET libraries provide the option of choosing from a large library of the existing components, which is much more reliable.
2. OPNET has been adopted by the several parts of the DOD and is used as a standard tool.
3. Since the library models have a large number of functionalities or parameters already implemented, programmers save a lot of time.

8. CONCLUSION

In this survey, we focused on the usage of OPNET in various scenarios ranging from simple simulations to verification of complex research ideas, deployment in mission critical applications, to a tool for optimization and extension. The advantages of our work are manifold, ranging from a review of state-of-art in network simulations, compilation of efforts being spent in the area. This survey will be greatly helpful for beginners as well as experienced users to give them a good introduction to the versatility of the tool, the type of applications that OPNET is used for by the researchers and the limitations of the existing OPNET version. It also evaluates OPNET vis-a-vis other commonly used simulators.

REFERENCES

[1] Opnet technologies, www.opnet.com

[2] S. Mittal et al., "Design Exploration and Implementation of Simplex Algorithm over Reconfigurable Computing Platforms," *IEEE International Conference on Digital Convergence*, pp. 204-209, 2011.

[3] S. Gupta et el., "Guaranteed QoS with MIMO Systems for Scalable Low Motion Video Streaming Over Scarce Resource Wireless Channels," *Proceedings of Second International Conference on Information Processing, I.K. International Pvt Ltd*, pp. 452-466, 2008.

[4] S. Mittal , S. Gupta and S. Dasgupta, "System Generator: The State-Of-Art FPGA Design Tool For DSP Applications", *Third International Innovative Conference On Embedded Systems, Mobile Communication And Computing (ICEMC2 2008)*, Global Education Center, India, 2008.

[5] X. Chang, I999. Network simulations with OPNET. In *Proceedings of the 31st Conference on Winter Simulation: Simulation—A Bridge To the Future* – Vol.1 (Phoenix, Arizona, United States, December 05 - 08, 1999.

[6] V.Y. Hnatyshin, and A.F. Lobo, 2008. *Undergraduate data communications and networking projects using opnet and wireshark software. In Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education* (Portland, OR, USA. SIGCSE '08. ACM, New York, NY, pp. 241-245, March 2008.

[7] J. Guo, W. Xiang, S. Wang, "Reinforce Networking Theory with OPNET Simulation", *Journal of Information Technology Education,* vol. 6, pp. 215-226, 2007.

[8] Computing Curriculum. (2001). *IEEE/ACM CC-2001 Task Force*. http://www.acm.org/sigcse/cc2001/index.html

[9] D. Comer, *Hands-on networking with internet technologies* (2nd ed.). Prentice Hall, 2004.

[10] http://www.lmu.ac.uk/ies/comp/esp lmu.html.

[11] http://www.eecs.umich.edu/dco/docs/ecad/opnet docs/UseOpnet.pdf.

[12] J.A. Zubairi, and M. Zuber, "Suny Fredonia Campus Network Simulation and Performance Analysis using OPNET", in *OPNETWORK2000*, Aug 2000, Washington DC.

[13] A.T. Koc, O. Ozdemir, M. Torlak, and W.P. Osborne, "Opnet simulation of Random Access CDMA Networks", in *OPNETWORK* 2002.

[14] G. F. Lucio, M. Paredes-Farrera, E. F. Jammeh, and M. J. Reed, "Opnet modeler and ns-2 - comparing the accuracy of network simulators for packet-level analysis using a network testbed." *WSEAS Transactions on Computers*, vol. 2, no. 3, pp. 700-707, July 2003.

[15] S. Ba ja j, L. Breslau, D. Estrin, K. Fall,S. Floyd, P. Haldar, M. Handley, A. Helmy,J. Heidemann, P. Huang, S. Kumar, S. McCanne,R. Rejaie, P. Sharma, K. Varadhan, Y.X.,Haobo Yu, D. Zappal "Improving Simulation for Network Research", *Technical Report, USC Computer Science Department*, March 1999.

[16] M. Zhou, S.D. Lang, "A Frequency-Based Approach to Intrusion Detection", in *Proceedings of the Workshop on Network Security Threats and Countermeasures*, July 2003.

[17] N. Alborz, M. Keyvani, M. Nikolic,and L. Trajkovic,, *Simulation of packet data networks using OPNET*, in OPNETWORK, 2000.

[18] G. Niestegge, "The leaky bucket policing method in asynchronous transfer mode networks*", International Journal of Digital and Analog Communication Systems*, vol. 3, pp. 187-197, 1990.

[19] T. Park, A. Dadej, "OPNET Simulation Modeling and Analysis of Enhanced Mobile IP" in *IEEE Wireless Communications and Networking Conference,* vol 2, pp 1017-1024, 2003.

[20] M. Buettner and D. Grunwald, "A Wireless Flit-Based OpNET Model", *Technical Report, Department of Computer Science*, University of Colorado at Boulder, May 2006.

[21] W. J. Dally and C. L. Seitz. "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Transactions on Computers*, Vol. 36, No. 5, pp. 547-553, May 1987

[22] N. Davis, S. Ransbottom, and D. Hamilton, 1998. *Teaching computer networks through modeling*. Ada Lett. XVIII, 5, 104-110, Sept. 1998.

[23] X.Yu, "ATM Network Simulation With OPNET", *Project Report on Communication Networks, Department of Electrical and Computer Engineering*, Michigan Technological University.

[24] M. Dooley, J. Dallaire and J. Reaper, "Integrating. Behavioral Models with Detailed OPNET Network Models. In a Distributed Framework", *International Symposium on. Collaborative Technologies and Systems*, 2003.

[25] J. Wang. and K. Nahrstedt 2000 *Design and Implement Differentiated Service Routers in OPNET. Technical Report*. UMI Order Number: UIUCDCS-R-2000-2177., University of Illinois at Urbana-Champaign.

[26] S.Shenker R. Braden, D. Clark, S. Shenker *Integrated Services in the Internet Architecture: an Overview*. RFC 1633, June 1994.

[27] S.Blake S. Blake, D.Black, M. Carlson, E. Davies, Z.Wang, W. Weiss *An Architecture for Differentiated Services*. RFC 2475, December 1998.

[28] A.Y. Lu, D. Liang, "OPNET Optimization of Ad-hoc sensor network configuration for navigation and positioning," *2005 IEEE Intemational Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings*, 2005.

[29] S. Gupta et al., "MIMO Systems For Ensuring Multimedia QoS Over Scarce Resource Wireless Networks," *ACM International Conference On Advance Computing, India*, 2008.

[30] S. Mittal, S. Gupta. "FPGA: An Efficient And Promising Platform For Real-Time Image Processing Applications," *National Conference On Research Development In Hardware Systems (CSI-RDHS)*, Kolkata India 2008.

ABOUT THE AUTHOR

Sparsh Mittal. Sparsh Mittal received his BTech in Electronics and Communications Engineering from the Indian Institute of Technology, Roorkee, India. He was the graduating topper of his batch and also received best B. Tech project award. Currently, he is pursuing his PhD in Electrical and Computer Engineering at Iowa State University (ISU), USA. He has been awarded scholarship and fellowship from IIT Roorkee and ISU. He has been a reviewer of conferences like IEEE Intelligent Systems, IEEE ISVLSI (IEEE Annual Symposium on VLSI), IEEE ISWTA (IEEE Symposium on Wireless Telecommunications Applications), and book series such as Understanding Complex Systems (Springer) etc. He has published more than 10 papers in conferences and journals. His research interests include natural language processing, data mining and FPGA implementation using system generator and VHDL.