ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Packet Load Generator for Telecom Networks

Niranjan U, Ramacharyulu P Govardhanam, K Vijaya

Abstract

In this is an era of drastic evolution Telecommunications, packet data communication has gained paramount attention due to increased internet multimedia usage. One of the challenges faced by telecom companies is to provide robust and powerful network components that are capable to handle the tremendous load of traffic and data on it. Companies evaluate their products performance before releasing them to the market by applying a large amount of generated traffic in order to measure their capability under traffic load; powerful solutions are hence needed for generating traffic and modeling various telecom protocols. The performance is also need to be evaluated for other scenarios that exists in the real network. The scenarios include capacity tests, busy hour call loads, rainy day scenarios, and different types of handoffs. Hence there is a requirement of a simulator tool which evaluates the product's performance. Packet Load Generator (PLG) is a tool developed to evaluate Network Controller.

***Index Terms*— Network Controller, RF path loss, Semi- marcovian Traffic, Traffic Models.**

I. INTRODUCTION

The ever increasing need for the multimedia communication over telecom network is guiding the telecom industry to better the existing features and data rates. Thus developed new or enhanced technologies cannot be introduced directly to the public. The expected or the theoretical figures from the network elements have to be tested rigorously for their capacity before it is deployed. The vendors cannot test and verify the figures and functionalities with the existing real world network. So there is a need for the network simulators that imitate the actual traffic scenarios and hence validate the newly developed network components.

Network components being complex entities challenge building such simulator tools. The telecom network components are majorly IP packet networks and thus such simulation tool can be equally employed for all kinds of IP networks for capacity and functionality testing or the existing packet generator tool can be employed with new functionalities embedded.

The tool has to consider the intermediate protocols and interfaces lying in connecting such network components. It have to include the air interfaces between switching center to Access Terminals, the Backhaul interfaces and protocols with the main network element and the IP link with the PDSN (Packet Data Serving Node) for internet connections. Major parameters of the network components determined will be the delays and losses of the packets transmitted.

Major challenge for the tool design is to create a virtual environment almost similar to the actual 3G network. Functionalities like Idle, Soft, Softer, Virtual Soft and Virtual Softer Handoff's are to be simulated between the tool simulated cells with the handoff's specified between which particular cells or sectors clearly. Measurements like Busy Hour Call Attempts( ), Connection Disconnects(connection and disconnection requests per hour to a particular cell) are to be simulated which are generally the mathematical considerations made from the times of Public telephone lines capacity analysis. Traffic is to be measured on both the sides Reverse Link (RL) and Forward Link (FL) and are measured in kilobitspersecond(kbps) and displayed to the user so that he can notice the rise and drops in the traffic with the required functionality and required number of cells simulated.

Many mathematical models are suggested for the figures used in generating the random numbers used in any of the steps here in the simulation. Generally the number of requests are said to be following Poisson distribution and some of the parameters like traffic are said to follow Erlang's distribution curve. Also Gaussian distribution functions holds similar to the Register and Deregistering the calls.

Different people have put forth their ideas of different aspects of this tool but a technical paper that enumerates all the conceptual requisites of such a tool is hardly available. This was our motive and this paper describes the key requirements of building an efficient simulation tool that exactly imitates the complex telecom network components and hence aids for capacity and performance testing. The traffic generation procedure, proposed in this paper, is highly modular. The proposed traffic generator architecture must be flexible and capable of producing traffic streams compliant with fully general high-level traffic models, either stochastic or deterministic. The software that drives the traffic generation process should employ several kinds of optimization, for removing limitations due to hardware speed and memory constraints of the host machine, and for ensuring that traffic streams corresponding to models with a wide range of parameters can be generated in an efficient and reliable manner. This tool can be used in conjunction with the traffic generator to support measurement-based performance evaluation in experimental test beds.

We give an example of the *state of art* of such a tool for the CDMA EVDO case and same techniques or architecture can be followed for UMTS networks with slight protocol and interface changes.

This paper first gives an overview of a EVDO network under next heading. The next heading deals with theory behind modeling the internet traffic for data used in data communication, then the basic architecture of the state space model is described, later we discuss about the importance of physical layer considerations in UMTS network and at the end, we give a over view of how the tool architecture.

II. INTERNET TRAFFIC THEORY

Observations of traffic on network links typically reveal intensity levels (in bits/sec) averaged over periods of 5 to 10 minutes which are relatively predictable from day to day. Systematic intensity variations occur within the day reflecting user activity. It is possible to detect a busy period (usually in the afternoon between 2 and 5 pm) during which the traffic intensity is roughly constant. This constancy suggests that Internet traffic, like telephone traffic, can be modeled as a stationary stochastic process here statistical variations occur about an underlying constant intensity. Busy period performance is then estimated by the long term average behavior derived for the stationary process.[2]

But still there is some ambiguity in the literature that whether the statistical theory of public telephone network can be applied as it is to the TCP or UDP supported internet traffic systems. The precise characteristics of this stationary process depend on the composition of Internet traffic. Currently, some 90 to 95% of Internet packets use TCP and correspond to the transfer of digital documents of one form or another (Web pages, data files, MP3 tracks …). The congestion avoidance algorithms of TCP cause throughput to vary elastically in reaction to random changes in the set of transfers in progress. A small but growing proportion of traffic relates to inelastic streaming audio and video transmission for both interactive and playback applications.

A. Traffic Objects

The traffic process can be described in terms of the characteristics of a number of objects, including packets, bursts, flows, sessions and connections, depending on the time scale of relevant statistical variations. The preferred choice for modeling purposes depends on the object to which traffic controls are applied. Conversely, in designing traffic controls it is necessary to bear in mind the facility of characterizing the implied traffic object. This consideration is particularly important in the design of the future Internet where only the datagram and the broad destination-based aggregate used in routing are currently recognized. Traffic characterization proves most convenient at an intermediate *flow* level.

A flow is defined for present purposes as the unidirectional succession of packets relating to one instance of an application (sometimes referred to as a microflow). For practical purposes, the packets belonging to a given flow have the same identifier (e.g., source and destination addresses and port numbers) and occur with a maximum separation of a few seconds. Flows are frequently emitted successively and in parallel in what are loosely termed "*sessions*." A session corresponds to a continuous period of activity during which a user generates a set of elastic or streaming flows.

Some network service models define the notion of "*connection*" and control resource allocation by means of explicit signaling exchanges. The connection might be set up for a particular flow or used over a long period for an aggregation of flows between given network end points. A significant difficulty resides in defining parsimonious traffic descriptors representing the impact the connection is likely to have on network performance. Signaling overhead may also prove excessive, particularly when each connection relates to an individual flow.

B. Arrival Processes And Service Requirements

It is well known that the arrival process of IP packets can exhibit extreme rate variations at multiple time scales. First reports of this behavior more than ten years ago have given rise to a large amount of research aiming to explain the so-called self similarity phenomenon and to predict its impact on network. The main reason for rate fluctuations at time scales greater than a few hundred milliseconds turns out to be extreme variability in the size of the flows making up the observed packet process. Yet more extreme variability (so called multi-fractal behavior) occurs at smaller time scales due to the burstiness induced by TCP. It proves much simpler to describe traffic in terms of flows.

The arrival process of flows in a backbone link typically results from the superposition of a large number of independent sessions. Observations confirm the predictable property that session arrivals can be assimilated to a Poisson process. This means simply that the probability of a new arrival in a short interval of length dt is equal to λdt, where λ is the arrival intensity, and is independent of all past activity. A Poisson process results naturally when traffic is due to the independent activity of a very large population of users, each individually having a very small intensity.

As a first approximation, it is not unreasonable to assume that individual flows also occur as a Poisson process. To ignore the correlation between flow arrivals within the same session is not necessarily significant when the number of sessions is large. It is also true that results derived under the simple Poisson assumption are also often true under more general assumptions.

III. TRAFFIC GENERATION THROUGH HIGH-LEVEL MODELS

A traffic model is one that have different states of the simulation tool and the advancements of the states are carried on according to *random process* that suits the particular physical process. Consider an example of Registering calls and states of tool is the number of calls registered per minute and this is modeled using Gaussian Process.

In [3], they discuss about the model for the system to change its state from one to another based on the traffic models and the present event.

A. The Structure of Semi-Markov Traffic Models

The traffic generator adopts a high level traffic specification in terms of (time invariant) semi- Markovian models. According to this framework, each traffic model is characterized by a set of N states. During a sojourn at some state i, a constant, state-dependent data rate ri is maintained. (The modeling framework assumes a "fluid-flow" approach, according to which more complex cell level rate fluctuations within each state can be ignored. This approach is entirely adequate for representing the important burst level phenomena, while at the same time avoiding excessive low-level complexity.) Transitions among states occur according to a discrete irreducible Markov chain with a transition probability matrix $P = [pij]$. The sojourns at a state i are independent and identically distributed random variables, of a general probability distribution function (PDF) $Fi(\cdot)$. The well-known MMRP traffic models, where all sojourn times are exponentially distributed, are an important special case of semi-Markovian models. [3]

B. Top Level Architecture

From a high level view, traffic generation corresponding to a semi-Markovian model consists of a sequence of events (high level traffic events—HLTEs). Each HLTE signifies the emission of cells at a constant rate *ri*, depending on the currently occupied state *i*, for a duration chosen through sampling the respective PDF governing the sojourns at this state.

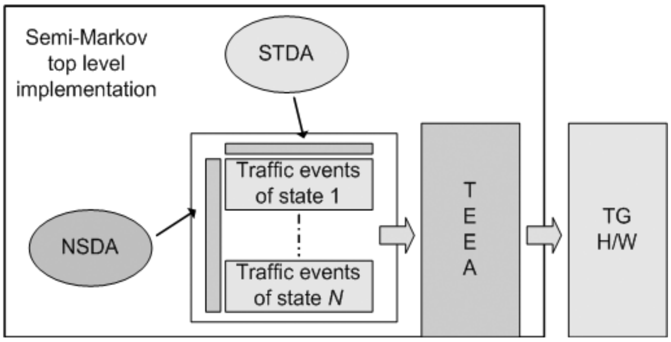

Fig.1. Basic functional diagram for generating semi-Markovian traffic.

With the notion of an HLTE at hand, the task of semi-Markovian traffic generation requires an algorithm to decide on the next state to be occupied by the model (next state decision algorithm—NSDA) and an algorithm to sample the sojourn time that will be spent at the chosen state (sojourn time decision algorithm—STDA). Furthermore, the nature of ATM and the structure of the traffic generation hardware suggest the need for two more mechanisms: One to convert data-rates, and the durations for which these rates are sustained, to ETEPs, i.e., (*ID*,*NC*) pairs (rate to pairs conversion algorithm—RPCA) and another to encapsulate the high-level traffic event sequence into units of size *HMS* (high level traffic events encapsulation algorithm—TEEA).The relation between these building blocks is depicted in Fig. 3.

C. The Next State Decision Algorithm (NSDA)

Given as input an Integer value expressing the state visited last time, the NSDA determines the state to be visited next. For this purpose, NSDA makes use of the transition probabilities matrix P = [*pij*], as follows: During an initialization step, performed once before the actual traffic generation process, NSDA computes the aggregate transition probabilities matrix A = [*aij*], where *aij* = _*j k*=0 *pik*, for all *i*, *j* = 0, ..., $N - 1$. During normal operation, given the state *i* currently occupied, the next state is decided upon by drawing a random number *s* in the interval [0, 1) and determining the smallest index *j*, such that the quantity *aij* exceeds *s*. Since the transition probability matrix *P* does not change over time, the partial probability sums are also invariant and having them ready into the elements of *A* reduces the time needed to determine the next state.

D. The Sojourn Time Decision Algorithm (STDA)

STDA consists of two modules: STDA 1, an initialization step performed once in startup, and STDA 2, which is executed repeatedly during the generation process. The preprocessing stage of STDA 1 undertakes the conversion of the sojourn PDFs into a form that allows for the efficient selection of actual sojourn samples by STDA 2 during on-line operation. Introduction of the preprocessing makes sense because the PDFs do not change over time. More specifically, STDA 1 reads data that determine the PDFs governing the sojourn times at all states of the model. Then, for each state the algorithm computes values of the respective PDF2 and creates an array of size *M* that contains inverse distribution function values (i.e., sojourn time values), which correspond to equally spaced PDF values in the range from 0 to $(M - 1)/M$. The dimension *M* (common to the arrays of all states) is chosen large enough to ensure a sufficiently dense representation of the PDF, while also avoiding excessive storage requirements. Once the arrays have been produced, STDA 1 exercises the RPCA algorithm upon each element of each array, taking as input the sojourn time of this element and the data rate associated with the corresponding state, and transforms this input into a collection of ETEPs. This is done in a way explained in the next subsection and further detailed in Appendix-A. During the cell generation process, the STDA 2 performs (in a highly efficient manner) a much simpler task, namely that of drawing a random index into the array corresponding to the state picked by the NSDA. The low-level traffic event information in the chosen element is then fed into TEEA, which has the responsibility of generating *HMS* ETEPs for download to the hardware. TEEA is described in Section III-F.

IV. PHYSICAL LAYER CONSIDERATIONS:

The advance communication theory proves that the RF multipath loss is less in the CDMA links compared to the TDM or FDM links due to its wide band nature, Hence physical layer consideration would not matter much in CDMA network simulation but is a point to be considered for UMTS networks.[4]

Most of the available commercial traffic generators utilize the network resources on the fixed side of the cellular network. This is a big disadvantage because such generators produce traffic without involving the overload of the air interface, which is the most important interface (Uu) at the UMTS network. In this way, we have not a complete and realistic view of the traffic generation, because it is well known that several congestion problems on 3G networks occur due to air interface limitations. The use of air interface resources, from a 3G traffic generator is very important, because CDMA networks like UMTS, have several crucial mechanisms that are implemented on RF layer, such as the open loop power control, admission control and fast close loop power control at both directions uplink/downlink. On the other hand, the most common use of a traffic generator is the ability to produce enormous traffic, in order to produce overload and congestion scenarios, for evaluating the response of the network in such situations.

In an UTRAN network the traffic generator, which works from the fixed network side will produce faulty and optimistic results comparing to traffic generator, which implements an "end to end" traffic generation. This means that on an UMTS traffic generator it is important to emulate not only the produced Erlang at the specific traffic scenario with determined services, but also to provide information about the response of the network to random and complex RF pathloss environment with multipath fadings etc.

Also, the majority of traffic generators need specific equipment and hardware, like terminals and control consoles, which is a minus point due to incompatibility. This proprietary architecture sometimes provides lack of scalability and of course increases the overall cost of the platform. The available traffic generators could be controlled only locally, which is not very practical. Also there is no capability to send the results and statistics remotely to a central database. [4]

In [5], a separate simulator tool design is discussed to imitate the RF losses in UMTS network.

Rayleigh fading generation

Generally in wireless transmission scenarios where a receiver is in motion relative to a transmitter with no line-of-sight path between their antennas, Rayleigh fading is often a good approximation of realistic channel conditions. The term *Rayleigh fading channel* refers to a multiplicative distortion *h(t)* of the transmitted signal *s(t)*, as in $\boldsymbol{y(t) = h(t) . s(t) + n(t)}$ where, *y(t)* is the received waveform and *n(t)* is the noise. The TGP is going to implement the fading generation, by performing a sampled version of the channel waveform *h(t)* in a statistically accurate and computationally efficient fashion. Thus, with the use of specific microwave filters and polyphase interpolators, we could produce such fadings on the Uu interface. But in our prototype we could emulate these fadings, with very fast alterations on the attenuation level on the programmable attenuators.

The channel waveform *h(t)* is modeled as a wide-sensestationary complex Gaussian process with zero mean, which makes the marginal distributions of the phase and amplitude at any given time uniform and Rayleigh, respectively, hence the term Rayleigh fading. The autocorrelation properties of the random process *h(t)* are governed by the Doppler frequency *fD*, as in[ ]:

$$R(\tau) = E\{h(t) \cdot h(t-\tau)\} \sim J_0(2\pi f_D \tau)$$

where *Jo*({) is the zero-order Bessel function of the first kind.

V. WORKING PROTOTYPE

Let us consider a CDMA EV-DO network as shown in the figure. Let us assume we are manufacturing an RNC and as explained earlier, it is really important to make sure of its capacity before rolling to down to the market. So, the tool that does this task needs all the requisites explained before. Let us consider we are using machines with any flavor of Unix operating system and works influenced by the before mentioned algorithms and design.

As we are analyzing an RNC, which is at the middle of the whole network, with network components on both sides, and packets sent from one side is received and sent to the components on the other side. This requires a simulation tool that should be capable of sending packets to the RNC and at the same time should be able to capture the packets from the other end and analyze it. Thus there is a need of a feedback type of architecture and complicates the design.

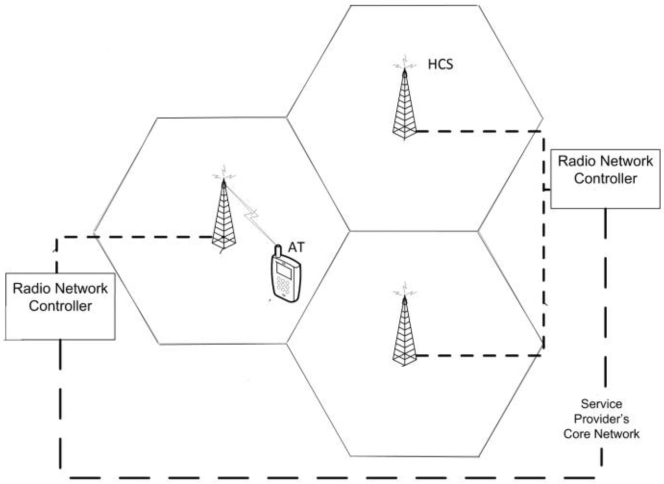

Figure 2. Basic block diagram of EV-DO network

The figure 3 gives a high level diagram of the possible simulation tool. One side, it acts as a generator and other side, analyzer. Some literature describes building two separate tools for these two purposes as Generator tool and Analyzer tool [1]. But extra complexities are to be faced for synchronizing both the tools and two separate architectures required for two different tools. In measuring delay and throughputs of the RNC under test, tools on both the sides must be matched in time and parameters. Small variations might lead to wrong measurements. Hence we, in this paper propose a single simulation tool that is capable of doing both the functions together. This includes the traffic model which is very near to the actual traffic scenarios in the existing networks. And the states changes are modeled by the next state algorithm referred earlier.

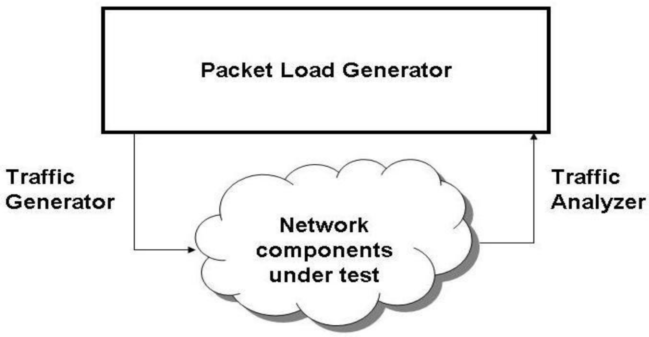

Figure 3. High level architecture

Since the traffic analyzer has to cope with the real time issues like computation of loss-oriented metrics and logging of results, there is a need for an autonomous entity that can work with the real time constraints. The performance of the traffic analyzer depends on the reduction in time consumption in processing the packets in real time. The module has to pass system calls each time a packet has to be processed. Instead of this, if the control is passed to the Unix kernel space and the packet processing module be implemented to work in the kernel space, then the overhead of system calls can be eliminated. Hence UNIX kernel space will be the best choice for the intended purpose.

The interfaces of the tool with other network components like HCS (HDR Cell Site) or the link with PDSN takes place in different interfacing protocols in different levels on the protocol stack. Hence the packets destined to the different components are to be prepared with the respective protocols and sent.

Apart from design and architecture, the simulation tool needs a good User Interface for the users to effectively and easily interact with the tool. Some literature deals with the geographical maps of urban or rural areas based on the RF path losses available from experimentation and GPS services are used to calculate the attenuation at each Access Terminal.[2]

Other useful information that that are desired by the user are the data rates on both the sides, time of the simulation and the number of calls registered and de-registered per hour basis. These data serve as the observations of the capacity testers and decisions are made based on these data.

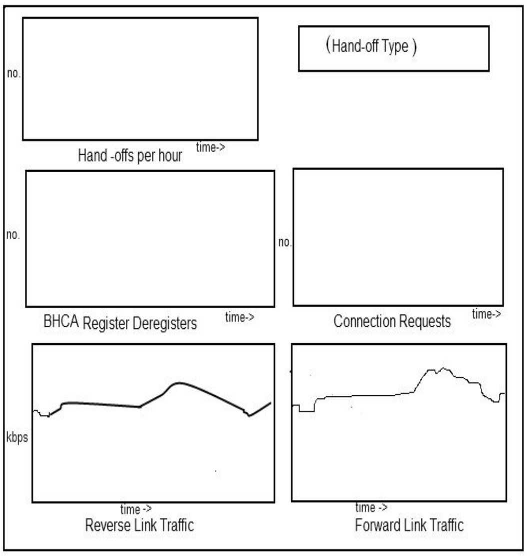

Figure 4 Prototype of User Activity window

The figure 4 gives an overview of a User window that gives the live details of some of the important details to the user. There will be a heartbeat time interval within which the tool gets the values from the network and refreshes the same on the window. Here Register deregisters are the number of Access Terminals registered and deregistered within the heartbeat interval. FL and RL are the link between Cell and AT and vice versa. The traffics on these links are also displayed. If the user is running the functionality of any different kind of hand-off, the number of hand-off's taken place per time interval and hand-off type is displayed. One of the main scenarios tested for telecom networks is Busy Hour Calling Attempts(BHCA) is also considered in the tool and the numbers are displayed on the user activity window. All these values are printed in their respective windows in a line joining points pattern and user can see the up's and down's occurred across the simulation period. Decreasing the heartbeat time gives more precise updates and come with disadvantage of consuming more processor and memory resources.

VI. CONCLUSION

In this paper we have presented a prototype traffic generator that can be used in CDMA networks and gives the ability operators to validate their networks. This is very important tool for the companies working on development and enhancements of telecom network components. The paper does not include any results, as the system cannot be compared with similar one. However the systems are build on such architecture and is used widely.

The system concept is simple; though innovative and can strongly support the operator in its planning and acceptance needs. In the future the prototype will become a commercial product, including other air-interface modules as well.

ACKNOWLEDGMENT

I am thankful to Alcatel-Lucent, Bangalore and my manager, Ramacharyulu GP and guide Mrs. K Vijaya, for their constant encouragement, guidance and support since beginning, which has transformed me as a "Subject Matter Expert".

Last but not least, I would like to extend my gratitude to all those who indirectly supported me for the success of this paper.

REFERENCES

[1] Loukatos, L. Sarakis, K. Kontovasilis, C. Skianis, G. Kormentzas, "Tools And Practices For Measurement–Based Network Performance Evaluation", The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07), 2007.

[2] Jim W. Roberts, France Telecom R&D, "Traffic Theory and the Internet", IEEE Communications Magazine, 2001.

[3] Dimitrios Loukatos, Lambros Sarakis, Kimon Kontovasilis, and Nikolas Mitrou ,"An Efficient ATM Traffic Generator for the Real-Time Production of a Large Class of Complex Traffic Profiles", Journal of communications and networks, vol. 7, no. 1, march 2005.

[4] Nikolaos Chr. Papaoulakis, Maria Marchitti, Sofoklis Kyriazakos,Neeli Prasad, "Traffic Generation Platform in W-CDMA Networks", IEEE 2007.

[5] Fratu, S Halunga, D N Vurreanu, "Radio Link Simulation Method For Umts Communications In Rayleigh Fading Environment", IEEE 2003.

Niranjan U, presently pursuing M.Tech in Digital Communication at BMSCE, Bangalore. I did B.E in Telecommunications from BNMIT, Bangalore and published a paper earlier on Speech encoding in IJCA. Presently I am doing internship in Alcatel Lucent in a tools development team for EVDO technology.

Mr. Ramacharyulu GP is presently Technical Manages in Alcatel-Lucent India Limited, Bangalore in a DO tools development team. He completed his B.Tech in Andhra University and M.S. in BITS, Pilani. He has more than 25 years of experience. He have worked on various telecommunication projects in companies like Ericson, HP, ALU on different Simulation tools for Wireless, 2G, 3G technologies.

Mrs. K Vijaya is currently Assistant professor at BMS College of Engineering, Bangalore. She has done B.E and M.Tech under Electronics and Communications. She has 15 years of industrial experience and 6 years of teaching experience. She has guided many M.Tech and B.E level projects and published many technical papers.