Case Study

International Journal of Advanced Computer Research, Vol 9(42)
ISSN (Print): 2249-7277 ISSN (Online): 2277-7970
http://dx.doi.org/10.19101/IJACR.2019.940029

A comprehensive study of process calculi with routing tables

Priyanka Gupta[1*] and Manish Gaur[2]
Research Scholar, Department of Computer Science, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India[1]
Professor, Department of Computer Science, Dr. A.P.J. Abdul Kalam Technical University, Lucknow, India[2]

Received: 30-January-2019; Revised: 06-April-2019; Accepted: 10-April-2019

Abstract
In this paper, we have described the research carried out in formal modeling of distributed networks in a process algebraic framework. The extended version of asynchronous distributed π-calculus named as routing calculi, DR_π^ω and DR_π were the one of the significant developments towards modeling the distributed computer network using the router as an active component and considering the path of communication between the communicating nodes where a routing table is a dynamic entity in a typical distributed network. In formal modeling, the routing tables was updated upon creation of the new node in the network, but already existing entries in the routing table remain unchanged. We have done a comprehensive study of previous year's research in this area on the basis of which we derive our motivation with an intention to extend the existing routing calculi DR_π^ω to incorporate the dynamic updates of the routing table through the distance routing protocol. It has adaptive features based upon the network parameter changes. This model is closer to the real networks. These calculi are primarily considered as a metric to determine the quality of services (QoS).

Keywords
Routing calculi, Process calculi, π-calculus, Routing protocols.

1.Introduction
With the evolution of large system over distributed computing environment, the need of verifying and validating the system becomes an important concern. The researches in theories of computing come up with the idea of process calculi like calculus of communicating system (CCS), communicating sequential processes (CSP), algebra of communicating processes (ACP) and language of temporal ordering specification (LOTOS). These are the major branches of the process calculi. Main research is started with Milner's work on the CCS [1] which is mainly based on observation bisimulation that defines communication between processed for the system. CSP by Hore's work [2] defines a specific calculus for defining a system and investigate to find correctness with its operational semantics and bisimulation. ACP [3] by Bergstra and Klop investigate the solutions of system equation, operational semantics and bisimulation. Finally, LOTOS [4], is a formal specification calculus, is used for protocol specification in ISO standards.

*Author for correspondence

Process calculi is a useful framework which is used in the study of concurrent and distributed system. One of the most popular calculus, the π-calculus [1, 5−8] of Milner, Parrow, and Walker is a simple and powerful calculus which describes the behavior of concurrent systems. A large amount of research work has been carried to define various application of process calculi in different system as the applied calculus [9] and the spi-calculus [10−12] have been used to study security protocol, the ambient calculus [13] defines mobile computations, the join calculus [14] has been used as basis for distribute implementations, the distributed π-calculus [15] for controlling the access to resources, the π_{cost} [16,17] defines cost function which calculates the cost of communication between the processes and the routing calculus [18] for distributing computing. Recently work on process calculi can be seen in [19−29].

We present an account of all these [30, 31] calculi to demonstrate both their features and limitation. Based upon the above discussion, we drive a motivation to develop a new routing calculi which can overcome one of the limitations of existing calculi to improve the QoS the formally described distributed network and then modeling this new development close to the real distribute network.

Further, this paper has been organized in six sections. In section 2, we describe related work for routing calculi in distributing computing. In section 3 comparative analyses has been described for process calculi. In section 4 and section 5, motivation and methodology have been discussed respectively. Finally, concluding section has been included in section 6.

2.Related work

Here in this section, we briefly describe some calculi that are relevant to the research problem. For this we describe asynchronous π-calculus, adding the cost in the π-calculus and routing calculus for distributed computing.

2.1Asynchronous π-calculus

Asynchronous π-calculus [5, 8, 32] is a computational and simple model for concurrent system [1]. In asynchronous π-calculus, the two concurrent processes can interact by sending values through communication links to each other i.e. where values can be interchanged between concurrent processes in a completely asynchronous manner through communication channels. These channels can be declared for the private use of a specific process. The syntax of the asynchronous π-calculus is as follows:

R :: = stop | u ! < v > | u? |(x) R | (*new n*) $R|R_1|R_2$

Here the simplest possible process, which does nothing, is stopped. The output of a value is represented by the term u! <v > that transmits the value v asynchronously via channel u. The term u?(x)R represents a term which can input a value along the channel u where x is the variable for the value to be communicated with the process R. If $v_1 = v_2$ then R_1 else R_2 is the test for the identity of simple values. (*new n*) R is used for scoping of names. R_1 | R_2 are running in parallel. Where R_1 and R_2 are two processes.

2.2D_π: Distributed π-calculus

The extension of π-calculus is asynchronous Distributed π-calculus [15] which is represented as D_π. Here the distributed system is viewed as a collection of domains, each capable of hosting computational processes, which in turn can migrate between domains. The topology of the communicating nodes is a clique of the graph which is not the case in the realistic distributed network.

In distributed π-calculus system is defined as:

$l\,[\![p_1]\!]\ (new\ e{:}\,D)\ (\mathrm{k}[\![Q]\!] \mid l\,[\![P_2]\!])$

Here, these are three located processes or agents where P_1 is running at location l, Q is running at location k and P_2 is running at location l. e is common resource which is a common sharing of type *D*. The new addition to the syntax of processes of this language is a migration which is represented as *goto k*. P meaning the running at any location l with migrate to location k where the code P is launched. Others are channel name creation and new location name creation. In this language, there are not a direct connection between this formal syntax for the system by using parallel contractor –|– and new name constructor (new e: D) (–). The axioms are inherited from the reduction semantics for asynchronous π-calculus and now they are being applied to locate processes. The new migration rule [15] is represented as k $[\![goto\ l\,.\,P]\!] \rightarrow l[\![P]\!]$.

2.3π_{cost}: adding the cost in the π-calculus

π_{cost} [16, 17] is a simple language which is another variation of asynchronous π-calculus [5, 8, 32] where values can be exchanged between the concurrent processes in a completely asynchronous manner via communication channels that can be declared as private. The most important modification with respect to asynchronous π-calculus is a cost function which calculates the cost of communication between the processes in a distributed network (defined over channel names). The cost function is calculated with the number of hops as the message is passed between the nodes.

In π_{cost}, the syntactical representation of reduction semantic is $c \vdash P \rightarrow^n c \vdash Q$ where a process P reduces to another process Q. The cost of this reduction is n which is defined by the cost function c. $c \vdash P$ is defined as a system. Another most important modification with respect to asynchronous π-calculus is that the author has added a syntactic construct called delivery message i.e. *del (b, u)* which delivers a value *u* at channel *b* of a waiting input process. This is a delivery message which is produced by an output process.

2.4$DR^{\omega}_{\pi}, DR_{\pi}$: a routing calculus for distributed computing

Routing calculi [18] is a language that is a tertiary extension of typical asynchronous distributed π-calculus [15]. The network topology is defined as networks of routers considering router as a fixed entity. There are nodes that are connected to a router and the processes reside in nodes.

The extended version of asynchronous distributed π-calculus named as two routing calculi i.e. DR_{π}^{ω} [30] and DR_{π} [31] were the first significant development towards modeling the distributed compute network using router and considering the path between the communication nodes. Systems, nodes and processes are three main syntactic categories in the language.

This language has syntactic expressions of the form $\Gamma_c \rhd (R)n\ [P]$ Where Γ_c is a router connectivity, (R) is a fixed routing table which hosts not n where process P resides in it and executes reduction of processes with respect to fixed router connectivity (which in not a clique of graph). The path can be determined by the routing table where entries are dynamically updated upon creation of new nodes. Upon creation of new nodes, the author has described two methods of updating routing tables, one method is flooding [33] and the other method is backward learning updates [34].

We describe the routing calculi incorporating backward learning updates and flooding updates as follows:

1. DR_{π}^{ω}: Updating router via a routing calculus with backward learning updates

Upon the creation of new nodes, the author has described a method of updating routing table [30]. In backward learning, tables are updated only when the new node name in propagating message and delivery of values occurs. The routing tables determine the path communication along which the value to be communicated determine. The entries in the routing table are very much dependent on router connectivity. The main purpose of using this method is to determine the path. The routers maintain a table which can determine the next on the path towards the destination node at some router where the destination process runs. Every router has a routing table and is represented as (R) which consists of set of node names and its adjacent set of router names.

For example, in *Figure 1*, If a process at node a, which is router R_1 wants to communicate with a process at another node g which is at R_3 them router table (R_1) may return the adjacent router R_2 as next hop the communication path to g at R_3. This may be formally expressed as (R_1) (g) = R_2. Similarly, the next router R_2 may decide that the next router for communication with g is R_3. This may be determined using the routing table (R_2) at R_2 as an application of the routing table function. In another words (R_2) (g) = R_3. Author describes the notation *Ad j(R)* to represent the set of neighboring routers of R in Γ_c. Routing table is represented as:

Ad j(R) = R′|(R, R′|) ∈ Γ_c

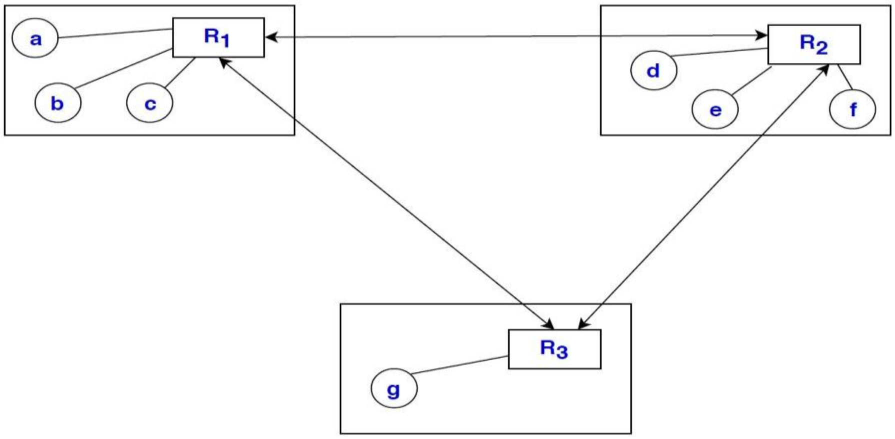

Figure 1 Sample distributed with routers and nodes

2. DR_{π}: Updating routers via routing calculus with flooding updates

Gaur et al. [31] describes that upon creation of a new node, a routing table is updated and information is broadcasted to all the connected routers and the routing tables are updated in updated in breadth-first traversal manner. Also generates two types of messages, firstly control message for updating routing table and secondly the value propagating message that is used to deliver value to waiting input process. With the help of the reduction semantics of this language, it can be confirmed that no router table is left without the knowledge about the new node after which the update messages get terminated. This method to update the routing table is known as flooding which is represented as DR_π [31].

3.Comparative analysis

The following table describes different parameters for showing the advantages and limitation of pervious and proposed calculi. It is also to consider that the following calculi are computational in nature.

Table 1 Comparison between process algebras

Process calculi	**Node representation**	**Connectivity**	**Routing table (Y/N)**	**Path changes**	**Node to not connectivity**
D_π	-	*goto* *k*.P	N	Static	Bidirectional
π_{cost}	-	*del* (b, u)	N	Static	Bidirectional
DR_π^ω	n [T]	Γ_c	Y	Static	Bidirectional
DR_π	m [T]	Γ_c	Y	Static	Bidirectional
Proposed calculi (DR_π)	-	Γ_c	Y	Static	Bidirectional

This table has been concluded in the next section.

4.Motivation

With the comparative analysis, which is given in *Table 1*, it is clear that the extended version of asynchronous distributed π-calculus [15] for routing calculi i.e. DR_π^ω [30] and DR_π [31] were the significant development towards modeling the distributed computer network using an active component named router. Considering the path of a communication between the communicating nodes where a routing table is a dynamic entity in a typical distributed network. After conducting an extensive survey, we discovered the two different calculi for routing table updates have considered static that has been explained in our previous section. In this calculus, the routing table is updated only when a new node is created which limited dynamic updates of the table. In formal modeling it is a constant endeavor of the researcher of formally describes it as close as possible to the real distributed network. With this motivation a research problem can be formulated. Therefore, we propose to develop a new calculi which will be an extension of routing calculi, DR_π^ω [30] to incorporate the following features and thereby overcoming limitation of previous calculi.

-Optimal route choices based upon dynamic parameters of the network [35].

-Dynamic update of routing tables to implement standard methods such as DVR [36, 37] and link state [38].

5.Methodology

We extend the existing calculi to incorporate the dynamic updates of routing table via distance vector routing [36, 37]. Distance Vector is a proactive routing protocol, which continuously maintains the topological information and such route information is available immediately whenever communication is needed. In distance vector router exchanges information about the network with their neighboring nodes by sending their routing table and router periodically transmits the routing table via a broadcast packet that reaches all other routers on the local segments. Thus, the router updates the routing table. Distance vector routing protocols use the Bellman-Ford algorithm [33] to compute the shortest route. Routing updates are done step by step from route to route in *Figure 2*.

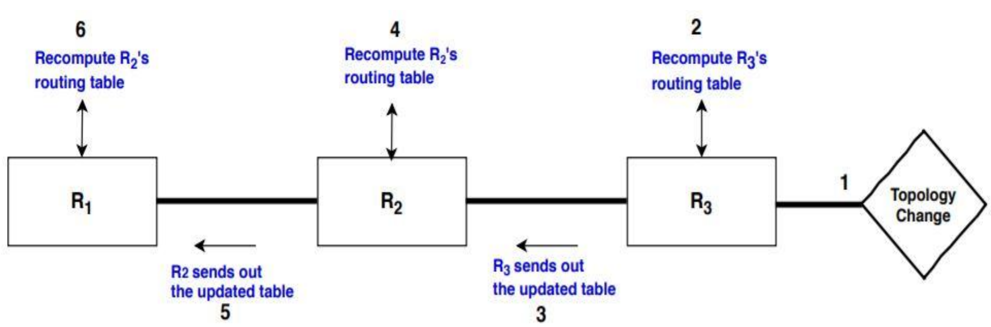

Figure 2 DVR: update routing information

Based on the limitation of previous work an extension of routing calculi with explicit features on routing table updates using a distance vector routing method have been developed [39]. This requires additional semantics rules. The challenges such as inconsistencies in simultaneous routing table are addressed. To ensure that all network properties are satisfied. We also define well-formed configurations. To justify our routing calculi, we take two routing calculi DR_{π}^{ω} [30] and DR_{π} [31] as its specification and show that both specifications conforms to its implementation which is our newly developed calculi. In *Figure 3*, we define the methodology of proposed calculus, which is stated as:

1. It shows reduction equivalence with DR_{π} (Distributed π-calculus) [15] abstracted away the unnecessary details of implementations [30].
2. Showing bisimulation equivalence over labelled transition system (LTS) [40] of the calculi to an equivalence defined purely in the observational properties of the calculi. (Full abstraction) [15, 30, 41]

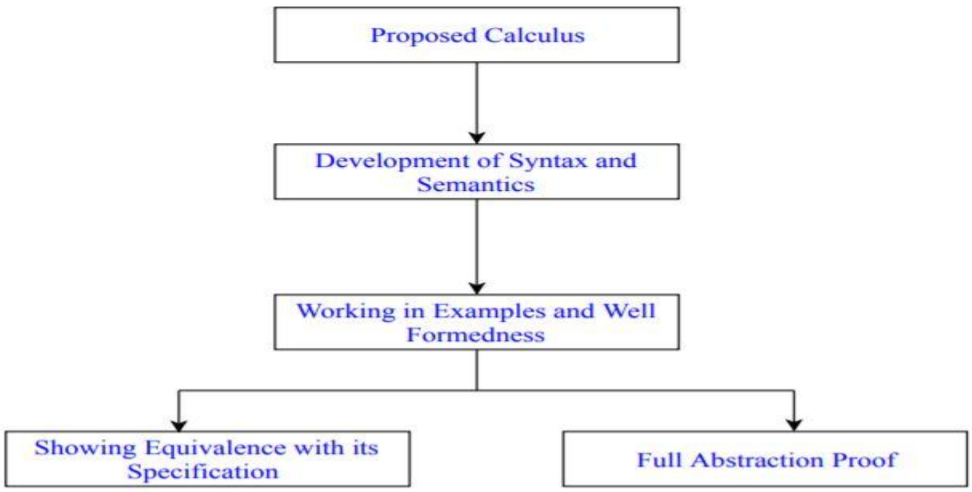

Figure 3 Methodology for proposed calculus

6.Conclusion

In this paper we have discussed various process calculi for routing in distributed system. Most of these researches had verified with the help of simulation tools such as a mobility workbench (MWB) [42], UPPAAL [43], simple promela interpreter (SPIN) [44] and LUNAR etc. The simulation tools had some limitation like qualitative aspects and other routing problems. Thus, the simulation tools could not be used to verify these systems by exploring all conditions related to them. Using formal methods, to overcome this situation, this system can be verified using theorem proven. That is why we can use the process algebraic framework. We observed in an existing routing calculus, the routing table (R) is a fixed table where as a routing table is a dynamic entity is a typical distribute network. In formal modeling, it is a constant endeavor of researcher to formally describe it as close as possible to the real distributed network. With this motivation, we extend the existing routing calculi to incorporate the dynamic updates of the routing table. As a prototype we can incorporate distance vector routing where the adjacent routes exchange their routing table to obtain an optimal path description. The syntax and semantics of the existing routing calculi are proposed in [39]. We seek to design an extension of the routing calculi to incorporate the features of distance routing protocol [36] which has an adaptive feature based upon the network parameter changes. Distance vector routing [36] has less computational complexity and message overhead.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

References

[1] Milner R. Communication and concurrency. New York etc. Prentice hall; 1989.

[2] Hoare CA. Communicating sequential processes. In the origin of concurrent programming 1978 (pp. 413-43). Springer, New York, NY.

[3] Bergstra JA, Klop JW. Algebra of communicating processes with abstraction. Theoretical Computer Science. 1985; 37:77-121.

[4] Turner KJ, Van Sinderen M. LOTOS specification style for OSI. In LOTOSphere: software development with LOTOS 1995 (pp. 137-59). Springer, Boston, MA.

[5] Milner R. Communicating and mobile systems: the π-calculus. Cambridge University Press; 1999.

[6] Milner R, Parrow J, Walker D. A calculus of mobile processes, I. Information and Computation. 1992; 100(1):1-40.

[7] Milner R, Parrow J, Walker D. A calculus of mobile processes, II. Information and Computation. 1992; 100(1):41-77.

[8] Parrow J. An introduction to the π-calculus. In Handbook of Process Algebra 2001:479-543. Elsevier Science.

[9] Abadi M, Blanchet B, Fournet C. The applied pi calculus: mobile values, new names, and secure communication. Journal of the ACM (JACM). 2018; 65(1):1-41.

[10] Abadi M, Gordon AD. A calculus for cryptographic protocols: the spi calculus. Information and Computation. 1999; 148(1):1-70.

[11] Lanotte R, Merro M, Munteanu A. A modest security analysis of cyber-physical systems: a case study. In international conference on formal techniques for distributed objects, components, and systems 2018 (pp. 58-78). Springer, Cham.

[12] Avalle M, Pironti A, Sisto R. Formal verification of security protocol implementations: a survey. Formal Aspects of Computing. 2014; 26(1):99-123.

[13] Bowman H, Derrick J. Formal methods for distributed processing: a survey of object-oriented approaches. Cambridge University Press; 2001.

[14] Edric Fournet C, Gonthier G. The reflexive CHAM and the join-calculus. In proceedings of the POPL 1996 (pp. 372-85).

[15] Hennessy M. A distributed Pi-calculus. Cambridge University Press; 2007.

[16] Hennessy M, Gaur M. Counting the cost in the picalculus. Electronic Notes in Theoretical Computer Science. 2009; 229(3):117-29.

[17] Guar M. A routing calculus towards formalising the cost of computation in a distributed computer network (Doctoral dissertation, University of Sussex).2007.

[18] Gaur M. A routing calculus for distributed computing. In proceedings of doctoral symposium held in conjunction with formal methods 2008 (pp. 23-32).

[19] Guillemin F, Mazumdar R, Rosenberg C, Ying Y. Network calculus for mean delay analysis through a network. In international teletraffic congress 2018 (pp. 55-60). IEEE.

[20] Yadav P, Gaur M. A behavioural theory for intrusion detection system in mobile ad-hoc networks. In proceedings of the international conference on high performance compilation, computing and communications 2018 (pp. 51-60). ACM.

[21] De Nicola R, Ferrari G, Pugliese R, Tiezzi F. A formal approach to the engineering of domain-specific distributed systems. In international conference on coordination languages and models 2018 (pp. 110-41). Springer, Cham.

[22] Audrito G, Viroli M, Damiani F, Pianini D, Beal J. A higher-order calculus of computational fields. ACM Transactions on Computational Logic. 2019; 20(1).

[23] Xie W, Zhu H, Wu X, Vinh PC. Formal verification of mCWQ using extended hoare logic. Mobile Networks and Applications. 2019; 24(1):134-44.

[24] De Nicola R, Gorla D, Pugliese R. Basic observables for a calculus for global computing. Information and Computation. 2007; 205(10):1491-525.

[25] Griffin TG, Sobrinho JL. Metarouting. In ACM SIGCOMM computer communication review 2005 (pp. 1-12). ACM.

[26] Sewell P, Wojciechowski PT, Pierce BC. Location-independent communication for mobile agents: a two-level architecture. In international conference on computer languages 1998 (pp. 1-31). Springer, Berlin, Heidelberg.

[27] Francalanza A, Hennessy M. A theory of system behaviour in the presence of node and link failures. In international conference on concurrency theory 2005 (pp. 368-82). Springer, Berlin, Heidelberg.

[28] De Nicola R, Ferrari GL, Pugliese R. KLAIM: a kernel language for agents interaction and mobility. IEEE Transactions on Software Engineering. 1998; 24(5):315-30.

[29] Montanari U, Sammartino M. Network-conscious π-calculus-a model of pastry. Electronic Notes in Theoretical Computer Science. 2015; 312:3-17.

[30] Gaur M, Kant R. A survey on process algebraic stochastic modelling of large distributed systems for its performance analysis. In international conference on eco-friendly computing and communication systems 2014 (pp. 206-11). IEEE.

[31] Gaur M, Gay SJ, Mackie I. A routing calculus with flooding updates. In international conference on distributed computing and internet technology 2015 (pp. 181-6). Springer, Cham.

[32] Sangiorgi D, Walker D. The pi-calculus: a theory of mobile processes. Cambridge University Press; 2003.

[33] Authentication I. Service for computer networks. IEEE Communications Magazine. 1994; 163.

[34] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. MIT Press; 2009.

[35] Sobrinho JL. Network routing with path vector protocols: theory and applications. In proceedings of the conference on applications, technologies, architectures, and protocols for computer communications 2003 (pp. 49-60). ACM.

[36] Tang L, Liu Q. A survey on distance vector routing protocols. arXiv preprint arXiv:1111.1514.2011.

[37] Garcia-Luna-Aceves JJ. A unified approach to loop-free routing using distance vectors or link states. In

ACM SIGCOMM computer communication review 1989 (pp. 212-23). ACM.
[38] Valadas R. OSPF and IS-IS: from link state routing principles to technologies. CRC Press; 2019.
[39] Gupta P, Gaur M. A routing calculus with distance vector routing updates. International Journal of Advanced Computer Science and Applications. 2018; 9(10):556-64.
[40] Sangiorgi D. A theory of bisimulation for the π-calculus. ACTA Informatica. 1996; 33(1):69-97.
[41] Pierce BC, Sangiorgi D. Behavioral equivalence in the polymorphic pi-calculus. Journal of the ACM (JACM). 2000; 47(3):531-84.
[42] Victor B, Moller F. The mobility workbench-a tool for the π-calculus. In international conference on computer aided verification 1994 (pp. 428-40). Springer, Berlin, Heidelberg.
[43] Bengtsson J, Larsen KG, Larsson F, Pettersson P, Yi W. UPPAAL in 1995. In international workshop on tools and algorithms for the construction and analysis of systems 1996 (pp. 431-4). Springer, Berlin, Heidelberg.
[44] D'Souza KA, Khator SK. A survey of petri net applications in modeling controls for automated manufacturing systems. Computers in Industry. 1994; 24(1):5-16.

Priyanka Gupta is a Ph.D. Research Scholar. Her research area is Formal Methods and Process Calculi. Before commencing her doctorate studies at the Dr. A.P.J. Abdul Kalam Technical University, Priyanka worked in the Computer Science Department at the Institute of Engineering and Technology Lucknow.
Email: priyankagupta1409@gmail.com

Dr. Manish Gaur is a Professor in Computer Science at Institute of Engineering and Technology, Lucknow (a constituent college of Dr APJ Abdul Kalam University Lucknow (AKTU)) and is presently working as Director of Center for Advanced Studies, AKTU. His research area can be broadly classified as Formal Methods and Verification of Large Systems; Semantics of Concurrent Programming Languages. His recent work interest includes formal modelling of stochastic behaviour of processes in a distributed network. Prof. Gaur is an Honorary Research Fellow of University of Glasgow, UK. Prof. Gaur obtained his doctorate degree in computer science from the University of Sussex, UK; M.Tech. in Computer Science from IIT Delhi and Bachelor in Engineering from NIT, Surat. Prof. Gaur’s academic track has been extraordinary throughout. He is a recipient of Commonwealth Academic Fellowship in 2012 and Commonwealth Scholarship in 2005 from the Commonwealth Scholarship Commission UK. Dr. Gaur was awarded the Young Scientist Award of Government of Uttar Pradesh in the Year 2003 for his contributions to the academics and research. His M.Tech. project was awarded the best project during the academic year 2001-02 at the department of Computer Science and Engineering at IIT Delhi leading to the conferment of prestigious Suresh Chandra Memorial Award. Dr. Gaur has 22 research papers to his credit. All his publications are in widely acclaimed journals and conference proceedings. Many consultancy and infrastructure projects from reputed organizations such as Google, Uttar Pradesh Subordinate Service Commission (UPSSC) are in his credit. He has held various academic and administrative positions at the university. He has widely travelled both within India and abroad, primarily in Western Europe and Middle East, for various academic purposes such as invited talks, conferences, short courses and project collaborations. He is working as founder director of the newly created institution, Center for Advanced Studies at AKTU since July 2017. The center has come up with state-of-the-art laboratories, post graduate academic programs in the field of Cyber Security, Mechtronics and Nano Technology under his leadership.
Email: manish.guar@ietlucknow.ac.in