A Stochastic Extension of the Routing Calculi

Rama Kant
Computer Science and Engineering Department
Institute of Engineering and Technology
Lucknow, India

Manish Gaur
Computer Science and Engineering Department
Institute of Engineering and Technology
Lucknow, India

ABSTRACT

The modern distributed systems have not only functional requirements (i.e. absence of deadlock, livelock etc.) but also have non-functional requirements (i.e. security, reliability, performance, Quality of Service(QoS) etc.). The methods for checking their correctness and analyze their performance is at very primitive stage. In the last few decades, formal verification techniques such as process algebras offer a powerful and rigorous approach for establishing the correctness of computer systems. Routing calculi (a such process algebra which is an elaboration of asynchronous distributed Pi calculus) which models a distributed networks with router as an active component in determining the path between communicating processes. This algebra also take into account various types of routing tables updates upon creation of new nodes. The semantics of routing calculi has been defined to incorporate the cost of communicating processes after taking into consideration the number of routers crossings. In this paper, we survey to extend the routing calculi. This is done with an intention to aggregate the number of states in the state space of calculus. We propose this extension along the lines of PEPA nets. A brief sketch of the proposed extension is also given in this paper as future direction of our research.

General Terms

Pi Calculus, Formal Methods, PEPA nets, State Space Explosion

Keywords

Routing calculi, Stochastic Modelling, Performance Modelling, Aggregation

1. INTRODUCTION

Interactive systems (i.e. ATM, online-shopping , social sites, e-governance etc.) are permeating our everyday life now-a-days. They are increasing in size as well as in complexity every hour. There may be failures which are very expensive in terms of money, face value, time etc. Formal methods [41] can be adopted for checking the correctness and analyze the performance of modern distributed systems. Formal methods are an analytical approach relying on mathematical models.

The process algebra [37] is one of them which is an algebraic language for describing system behaviour. There are so many process algebras that have been proposed in the literature in order to model and analyze these distributed systems which are inherently concurrent. The classical process algebras such as [38, 34, 15, 48] were concerned with the functional aspects of the concurrent systems. One of the most popular process algebra is π -calculus[37]. Hundreds of its variants and extensions considering the qualitative aspects of various distributed systems are developed. Some of them are [23, 26, 24, 28, 22, 27, 21, 36, 25, 43].For example, π -calculus[37] is a model of computation for concurrent systems in which processes are defined as:

$$P ::= c!\langle v\rangle \mid c?(x)P \mid (new \quad n)P \mid P \mid P \mid stop$$

Where the simplest possible process, which does nothing, is represented by the term $stop$. The term $c!\langle v\rangle$ represents the next simplest process, which can transmit the value v along the channel c . Input from a channel c is represented by the term $c?(x)P$, where x is a variable which may be used to receive the incoming value. $P \mid P$ represents two processes running in parallel; they may exchange values using input/output on channels. $(new \quad n)P$ is a scoping mechanism for names.

Asynchronous π -Calculus [28] is one of the variations of π -calculus. This is a calculus describing the evolution and behaviour of asynchronous communication system. In this language values can be exchanged between concurrent processes via communication channels. Communication channels can be used to model resources. The syntax allows them to be declared as private, for the exclusive shared use of specific processes. The names of these channels/resources can also be transmitted between processes.

Asynchronous π -calculus [28] is further extended in [23, 24, 26, 12] to model a distributed network with routers acting as an active component in determining the quality of service [18] of the network. The two routing calculi, DR$_{\pi}^{\omega}$ and DR$_{\pi}$ [23, 12], are developed with the intention of modelling a distributed network to demonstrate the cost of communication between the communicating processes. The cost of communication is the number of hops (router) a value propagating message crosses before delivering it to the destination in a network of routers. In fact the value propagating messages used in these models closely resemble the IP packet in TCP/IP model of networks [45]. However , the names (addresses) of source and destination nodes and data in messages are used unlike real IP packets where lots of other information are contained in it.

The syntax of the DR$_{\pi}^{\omega}$ is given in Table 1. The processes in the DR$_{\pi}^{\omega}$ are π -calculus processes. The syntax of nodes is adopted from the asynchronous distributed Pi calculus [28]. The new syntactic category in the DR$_{\pi}^{\omega}$ is the system. A system is represented by $\langle R\rangle M$, where R is a router and M is another syntactic category named as nodes. The nodes in M are directly connected to the router R. Two parallel systems are represented by S │ T and [R]M$_{sg}^{k}$ (n,m,v@c) is a message at router R. This message is used to propagate the value v from one router to another router during communication between some process at source node n to another process at destination node m. The value propagated by the message is represented by v@c to deliver value v to the specified channel c of the destination process. The integer k indicates the number of hops (routers) the message has already travelled across the path towards its destination. (new d)S is a scoping mechanism for names.

Further, asynchronous π-calculus [28] is also extended to π_{cost} [24, 23]. Where, the π_{cost} [24, 23] incorporates the cost of a π-calculus computation in some cost framework. This framework is based on the type setting of typed asynchronous π-calculus [27]. The syntax of π_{cost} is given in Table 2. It is a minor extension of the π-calculus and self explanatory.

In all these algebras time is abstracted away as they are developed for qualitative analysis rather than quantitative. Stochastic extensions of process algebras [33,2] are developed to add quantification to process algebra models. This has been done by the use of a random variable which has some probability distribution. For example, PEPA (Performance Evaluation Process Algebra) [33] is developed to investigate how the compositional features of the process algebra might impact upon the practice of performance modelling. It extends the classical process algebra [37, 38] by associating a random variable with every action for representing duration. These random variables are assumed to be exponentially distributed [47]. This leads to a clear relationship between the process algebra model and a Continuous Time Markov Chains (CTMC) [47]. Continuous Time Markov Chains (CTMC) are used to obtain performance measures with standard numerical techniques. PEPA has following syntax:

$$P ::= (a,\lambda).P \mid P+Q \mid P \underset{L}{\bowtie} P \mid P/L \mid A$$

where $(a,\lambda).P$ denotes a component which may perform an activity (a,λ) of action type a with activity rate λ. The duration of each PEPA activity is determined by an associated exponential probability distribution function[47]. The probability distribution function is parameterized by the activity rate. $P+Q$ indicates that a component may behave as P or Q. $P \underset{L}{\bowtie} Q$ is the synchronization operator of PEPA. The components P and Q are required to synchronize over the action types in the set L and all the other actions are performed autonomously. P/L is used as a hiding operator. A is a constant label used to model cyclic behaviour.

Table 1: Syntax for DR$^{\omega}_{\pi}$

S,T::=	**Systems**
$\langle R\rangle M$	Router
S │ T	Concurrency
[R]M$^{k}_{sg}$ (n,m,v@c)	Messages
(new d)M	New Name
ε	Identity
M,N::=	**Nodes**
n[T]	Named Processes
M │ N	Concurrency
(new d)M	New Name
0	Identity
T,U::=	**Process Terms**
c?(x)T	Input
$m!\langle v@c\rangle$	Output
if u = v then T else U	Matching
(new b)T	Channel Name creation
newnode m with P in Q	New Node creation
T │ U	Concurrency
*T	Repetition
stop	Identity

Further, PEPA [33] is extended to PEPA nets [10,11] to model mobility in modern distributed systems. PEPA nets [10,11] uses the PEPA [33] as the inscription language for labelled stochastic Petri nets [1,50].

The existing process algebras for routing [23, 26, 24, 12] are either model the network with respect to the specific protocol or they are focused on cost effectiveness of the computations where the cost of communication is the number of hops (router) a value propagating message crosses before delivering it to the destination in a network of routers. None of these have discussed to address the state space explosion problem [49, 5] by aggregation technique [32]. A extensive survey regarding state space explosion problem can be found in [8].

Table 2: Syntax for π_{cost}

P,Q::=	**Process Terms**
u?(x).P	Provide resource u
u! $\langle v@c\rangle$ .P	Use resource u
if u = v then P else Q	Matching
subscribe(o,u,c).P	Subscribe to resource u
(new a:R)P	Resource creation
P │ Q	Concurrency

*P	Repetition
stop	Termination
del(a,v)	Asynchronous message delivery

The rest of the paper is organised as follows. In Section 2 we give problem statement. Section 3 describes the brief sketch of the solution. Section 4 is the conclusion.

2. PROBLEM STATEMENT

We propose an extension of routing calculi to model a distributed network of routers along the lines of PEPA nets [33]. A simple distributed network of routers shown in Figure 2, where router as an active component. The topology of the routers is fixed but choice of path is probabilistic. The processes reside in a located site called nodes. Nodes are directly connected to some specific router. Any two processes at nodes can communicate via the routers. The routers specify the paths, with some probability, across the network between the communicating processes. The processes communicate along these paths.

A typical system is described as, $\langle R\rangle n[P]$, in [23, 26, 12]. Where a process P is a PEPA [33] component. A process resides at a named node n which in turn is connected to router R. Each router, R, maintain a routing table represented by $\langle R\rangle$. Routing table, $\langle R\rangle$, determines the path between routers with some probability. This probability is based on different network properties (i.e. congestion, bandwidth, throughput delay, etc.). The reductions [23, 26, 12] are done on configuration, $\Gamma_C \rhd \langle R\rangle N$, where Γ_C is the network of routers connectivity. We assume that the routers connectivity, Γ_C, is fixed. We assume that the router connectivity is bidirectional and each pair of routers is connected via some path. Essentially the router connectivity is a connected undirected graph (but not a clique of a graph). So, there may exist more than one path between the same pair of routers. The routing table at each router determines which path be used for the communication between processes. The path with high probability will be chosen for communication between routers.

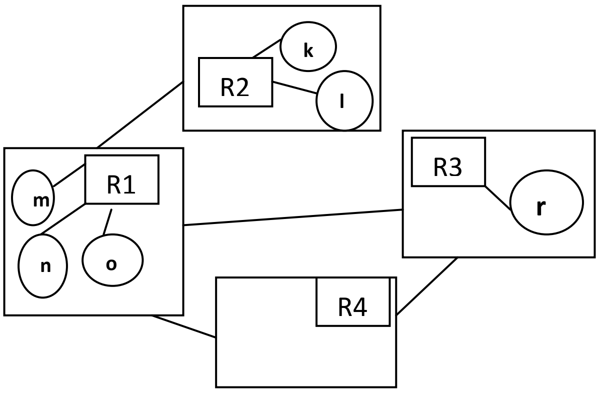

Figure 1: Example: A simple distributed network with routers

For example, in Figure 1, there are two paths between router R_1 and R_3. One path is direct and other is via router R_4. Let direct path has low probability then the path via router R_4. So the routing table $\langle R_1\rangle$ has two entries for the paths between router R_1 and router R_3 with their respective probability. Here router R_1 will choose the path for router R_3 which has high probability i.e the path via router R_4.

The syntax and semantics of our model will be developed along the line of PEPA nets [10]. We will aggregate the number of states in the state space of calculus along the lines of [13, 42, 8] to fight state space explosion problem. We intend to provide a formal proof based upon observational properties coinciding with bisimulation relation defined over a labelled transition system of the calculi. Thus making a language fully abstract. This method of proof is standard as described in [28, 37]. We will do qualitative as well as quantitative analysis of more realistic distributed network by the means of the proposed calculus.

3. BRIEF SKETCH OF SOLUTION

As we have proposed an extension of routing calculi, such as $\mathrm{DR}_{\pi}^{\omega}$ and DR_{π} [23, 12], along the lines of PEPA nets [10]. In routing calculi we need mobility of processes and values among the nodes as described in [23, 24, 12]. In our proposed extension, the processes will be extended along the lines of PEPA [33]. The mobility of processes will be extended along the lines of PEPA nets [10].

Further, Routing calculi is proven to be fully abstracted. This is an extension of the distributed Pi calculus [28]. The distributed Pi calculus [28] is an extension of the asynchronous Pi calculus[28] . The stochastic process algebra, PEPA [33], is also an extension of classical process algebra [28, 37] by introducing probabilistic branching and timing of transitions. The stochastic process algebra, PEPA [33], is well proven for functional and performance analysis. PEPA nets [10] is also an extension of stochastic process algebra, PEPA [33], allowing a number of distinct PEPA models to be arranged into a net. We will further progress our proposed work along this line of research according to following steps:

1. First we will extend the syntax and semantics of the routing calculi, $\mathrm{DR}_{\pi}^{\omega}$ and DR_{π} [23, 12], along the lines of [10]. We will also give well-formed conditions.
2. Further we will develop a formal proof based upon observational properties coinciding with bisimulation relation defined over a labelled transition system of the calculi.
3. If we will get positive results from the extension done in step 1 and 2 then we will extend the work by considering the probabilistic choice of paths.
4. We will also intend to provide an implementation with possibilities of plugging in it to PEPA Eclipse.

There is the considerable amount of literature available along this line of research in [32, 7, 30, 8, 11, 14, 1, 39, 9, 20, 46, 39, 19, 17, 16, 50, 29, 33, 31, 40, 35, 3, 44].

4. CONCLUSION

In this paper we propose and briefly outlined the future progress of the extension of the routing calculi [23, 26, 12] along the lines of [10]. A theoretical formulation based on process algebraic framework is intended to be developed along the lines of PEPA net and routing calculi.

A comparison of two modelling formalisms ,stochastic process algebra and stochastic Petri nets, in [6] concludes that “ there is scope for future work incorporating the attractive characteristics of the formalisms, such as structural analysis or functional abstraction, from one paradigm into the other ”.

There is strong intuition that our proposed extension of the routing calculi along the lines of [10] will give useful results for modelling more realistic distributed network.

5. REFERENCES

[1] Ajmone Marsan, M. and Bobbio, A. and Donatelli, S. Petri nets in performance analysis: An introduction. In Reisig, Wolfgang and Rozenberg, Grzegorz, editors, *Lectures on Petri Nets I: Basic Models* in Lecture Notes in Computer Science, pages 211-256. Springer Berlin Heidelberg, 1998.

[2] Marco Bernardo and Roberto Gorrieri and Mura Anteo Zamboni. A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time. *Theoretical Computer Science,* 202:1–54, 1997.

[3] Peter Buchholz. Markovian Process Algebra: Composition and Equivalence. pages 11–30, 1994.

[4] Chiola, G. and Dutheillet, C. and Franceschinis, G. and Haddad, S. Stochastic well-formed colored nets and symmetric modeling applications. *Computers, IEEE Transactions on*, 42(11):1343-1360, 1993.

[5] Clarke et. al. Model Checking and the State Explosion Problem. In Meyer, Bertrand and Nordio, Martin, editors, *Tools for Practical Software Verification* in Lecture Notes in Computer Science, pages 1-30. Springer Berlin Heidelberg, 2012.

[6] S. Donatelli and M. Ribaudo and J. Hillston. A Comparison of Performance Evaluation Process Algebra and Generalized Stochastic Petri Nets. *In Proc. 6th International Workshop on Petri Nets and Performance Models,* pages 158–168, 1995. IEEE Computer Society Press.

[7] Fuggetta, A. and Picco, G.P. and Vigna, Giovanni. Understanding code mobility. *Software Engineering, IEEE Transactions on*, 24(5):342-361, 1998.

[8] Manish Gaur and Rama Kant. A Survey on Process Algebraic Stochastic Modelling of Large Distributed Systems for its Performance Analysis (Accepted). *ICECCS* in to be appear in conference proceedings, 2014. IEEE Xplore Digital Library.

[9] Gilmore, Stephen and Hillston, Jane and Kloul, Lela and Ribaudo, Marina. Software Performance Modelling Using PEPA Nets. *Proceedings of the 4th International Workshop on Software and Performance* in WOSP ’04, pages 13–23, New York, NY, USA, 2004. ACM.

[10] Gilmore, Stephen and Hillston, Jane and Kloul, Leïla. PEPA Nets. In Calzarossa, MariaCarla and Gelenbe, Erol, editors, *Performance Tools and Applications to Networked Systems* in Lecture Notes in Computer Science, pages 311-335. Springer Berlin Heidelberg, 2004.

[11] Hillston, J and Ribaudo, M. Modelling Mobility with PEPA Nets.In Aykanat, Cevdet and Dayar, Tugrul and Korpeoglu, Ibrahim, editors, *ISCIS* in Lecture Notes in Computer Science, pages 513-522, 2004. Springer.

[12] Manish Gaur, Ian Mackie and Simon Gay. A Routing Calculus with Flooding Updates. *ICDCIT* in Lecture Notes in Computer Science, 2015. Springer.

[13] http://www.inf.ed.ac.uk/ teaching / courses /pm/ Note15.pdf.

[14] Allan Clark, Stephen Gilmore, Jane Hillston, and Mirco Tribastone. Stochastic Process Algebras. *LFCS, School of Informatics, University of Edinburgh*.

[15] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37(0):77 - 121, 1985.

[16] Brodo, Linda and Degano, Pierpaolo and Gilmore, Stephen and Hillston, Jane and Priami, Corrado. Performance Evaluation for Global Computation. In Priami, Corrado, editors, *Global Computing. Programming Environments, Languages, Security, and Analysis of Systems* in Lecture Notes in Computer Science, pages 229-253. Springer Berlin Heidelberg, 2003.

[17] Peter Buchholz. Hierarchical Markovian Models - Symmetries and Reduction-. *Performance Evaluation*, pages 234–246, 1992.

[18] Andrew T. Campbell and S. Keshav. Quality of service in distributed systems. *Computer Communications*, :21(4):291-293, 1998.

[19] Chiola, G. and Dutheillet, C. and Franceschinis, G. and Haddad, S. Stochastic Well-Formed Coloured Nets and Multiprocessor Modelling Applications. In Jensen, Kurt and Rozenberg, Grzegorz, editors, *High-level Petri Nets*, pages 504-530. Springer Berlin Heidelberg, 1991.

[20] Clark, Allan and Gilmore, Stephen and Hillston, Jane and Tribastone, Mirco. Stochastic Process Algebras. In Bernardo, Marco and Hillston, Jane, editors, *Formal Methods for Performance Evaluation* in Lecture Notes in Computer Science, pages 132-179. Springer Berlin Heidelberg, 2007.

[21] A. Francalanza and M. Hennessy. A theory of system behaviour in the presence of node and link failures. *In CONCUR*, :368-382, 2005.

[22] Adrian Francalanza and Matthew Hennessy. A theory of system behaviour in the presence of node and link failure. *Inf. Comput.*, :206(6):711-759, 2008.

[23] Manish Gaur. *A Routing Calculus: Towards formalising the cost of computation in a distributed computer network*. PhD, Informatics, University of Sussex, U.K., 2008.

[24] Manish Gaur and Matthew Hennessy. . Counting the cost in the picalculus (extended abstract). *Electronic Notes in Theoretical Computer Science (ENTCS)*, :229:117-129, 2009.

[25] Manish Gaur and Rama Kant. Article: $\mathrm{DR}_{\pi}^{\omega}\mathrm{F}$: A Fault Tolerant Distributed Routing Calculi. *International Journal of Computer Applications,* 40(7):1-7, 2012. Published by Foundation of Computer Science, New York, USA.

[26] Manish Gaur. A routing calculus for distributed computing. *In Elena Troubitsyna, editor, Proceedings of Doctoral Symposium held in conjunction with Formal*

Methods 2008,Turku Centre for Computer Science General Publication, 48:23-32., 2008.

[27] Matthew Hennessy and Julian Rathke. Typed behavioural equivalences for processes in thepresence of subtyping. *Mathematical Structures in Computer Science*, :14(5):651-684, 2004.

[28] Matthew Hennessy. *A distributed Pi-Calculus.*. Cambridge University Press, 2007.

[29] Holger Hermanns and Joost-Pieter Katoen. Automated Compositional Markov Chain Generation for a Plain-Old Telephone System. *SCIENCE OF COMPUTER PROGRAMMING*, pages 97–127, 1999.

[30] J .Hillston. Fluid flow approximation of PEPA models. *Quantitative Evaluation of Systems, 2005. Second International Conference on the*, pages 33-35, 2005.

[31] Hillston, Jane and Ciocchetta, Federica and Duguid, Adam and Gilmore, Stephen. Integrated Analysis from Abstract Stochastic Process Algebra Models. In Heiner, Monika and Uhrmacher, AdelindeM., editors, *Computational Methods in Systems Biology* in Lecture Notes in Computer Science, pages 2-4. Springer Berlin Heidelberg, 2008.

[32] Jane Hillston and Mirco Tribastone and Stephen Gilmore. Stochastic Process Algebras: From Individuals to Populations. *The Computer Journal*, 2011.

[33] Jane Hilston. A compositional Approach to Performance Modeling.Distinguished Dissertations in Computer Science. *Cambridge University Press.*, 1996.

[34] Hoare, C. A. R. Communicating Sequential Processes. *Commun. ACM*, 21(8):666–677, 1978.

[35] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *in "Conference Record of the 16th ACM Symposium on Principles of Programming Languages (POPL*, pages 344–352, 1989.

[36] M. Hennessy, J. Riely. Resource access control in systems of mobile agents. *Information and Computation .*, :173 :82-120, 2002.

[37] R. Milner. *Communicating and mobile systems: The π -Calculus*. Cambridge University Press, 1999.

[38] Robin Milner. A Calculus of Communicating Systems. *volume 92 of Lecture Notes in Computer Science. Springer*, 1980.

[39] Pedro R.D'Argenio, Joost-Pieter Katoen, and Ed Brinksma. An algebraic approach to the specification of stochastic systems(exteneded abstract). *In D.Gries and W.P.de Roever,editors, Proceedings of the IFIP working conference on Programming Concepts and Methods*, :126-147, 1998.

[40] Pourranjbar, Alireza and Hillston, Jane. An Aggregation Technique for Large-scale PEPA Models with Non-uniform Populations. *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools* in ValueTools '13, pages 20–29, ICST, Brussels, Belgium, Belgium, 2013. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[41] R.W.Butler. "What is Formal Methods". *Available online from http://shemesh.larc.nasa.gov/fm/fm-what.html.*

[42] Richard A. Hayden , Jeremy T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22 24):2260 2297, 2010.

[43] Rocco De Nicola, Daniele Gorla, and Rosario Pugliese. Basic observables for a calculus for global computing. *Inf. Comput.*, :205(10):1491-1525, 2007.

[44] Davide Sangiorgi and DavidWalker. *The π -Calculus: A theory of Mobile Processes.*. Cambridge University Press, 2001.

[45] W. Richard Stevens. *TCP/IP Illustrated : The Protocols,*, volume 1. Addison-Wesley, 1994.

[46] Tribastone, Mirco. The PEPA Plug-in Project. *Proceedings of the Fourth International Conference on Quantitative Evaluation of Systems* in QEST '07, pages 53–54, Washington, DC, USA, 2007. IEEE Computer Society.

[47] Trivedi, Kishor S. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley and Sons Ltd., Chichester, UK, 2nd edition edition, 2002.

[48] Kenneth J. Turner and Marten van Sinderen and K. J. Turner and M. Van Sinderen. LOTOS Specification Style for OSI. *The LOTOSPHERE Project*, pages 137–159, 1992. Kluwer.

[49] Valmari, Antti. The state explosion problem. In Reisig, Wolfgang and Rozenberg, Grzegorz, editors, *Lectures on Petri Nets I: Basic Models* in Lecture Notes in Computer Science, pages 429-528. Springer Berlin Heidelberg, 1998.

[50] W.Reisig. Petri Nets: An Introduction. *Springer-Verlag*, 1982.