A Routing Calculus with Distance Vector Routing Updates

Priyanka Gupta, Manish Gaur
Centre For Advanced Studies
Dr. A.P.J. Abdul Kalam Technical University
Lucknow, India

***Abstract*—We propose a routing calculus in a process algebraic framework to implement dynamic updates of routing table using distance vector routing. This calculus is an extension of an existing routing calculus DR_π^ω where routing tables are fixed except when new nodes are created in which case the routing tables are appended with relevant entries. The main objective of implementing dynamic routing updates is to demonstrate the formal modeling of distributed networks which is closer to the networks in practice. We justify our calculus by showing its reduction equivalence with its specification D_π (distributed π-calculus) after abstracting away the unnecessary details from our calculus which in fact is one of the implementations of D_π. We nomenclate our calculus with routing table updates as DR_π^φ.**

***Keywords*—Routing Calculi; Routing Protocols; Well Formed Configuration; Reduction Semantics**

I. Introduction

In recent years, developments in formal modeling of distributed networks in a process algebraic framework through process calculi has marked profound work [1], [2] [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. The extended version of Asynchronous Distributed π-calculus (AD_π) named as routing calculi DR_π^ω was one of the significant developments towards modeling the distributed computer network using an active component named router and considering the path of a communication between the communicating nodes where a routing table is a dynamic entity in a typical distributed network.

DR_π^ω consists of a network of routers of fixed topology. The processes reside in a located site called nodes which are directly connected to some specific router. Any two processes at nodes can communicate through the routers. The routers find the path over the network between the communicating processes. The processes communicate via this path.

A system in DR_π^ω, looks like $\Gamma_c \rhd S$. Here S can be of the form $\langle R \rangle [\![n[P]]\!]$ where P is a process that resides under node n connected at the same or a different node to some router R. The communication between the processes takes place through routers. Each process is located at some particular node which in turn resides at some particular router. The routers determine the particular path along the router connectivity through which the communicated values are forwarded. In this language, the routing table is updated only when a new node is created which limits dynamic updates of the table.

We present a new calculi DR_π^φ which is a direct adoption of routing calculus DR_π^ω [10] with a modified feature of routing table updates which is dynamic in DR_π^φ unlike DR_π^ω. We have abstracted away few details from DR_π^ω to demonstrate the power of new calculi in more simple way. These features can be adopted to DR_π^φ without much amendments to it. We describe a method for routing table updates with the help of implementation of distance vector routing method [13], [14], [15], [16] which uses the Bellman-Ford algorithm [17], [18], [19], [20] to compute the shortest route. In this calculi, the routing tables are periodically exchanged with their neighbors and with this updated entries new routes are found. We abstract away the details of new routes calculation methods by incorporating function δ in our semantics rule. Further to maintain the consistency in the calculi a clock $t_{k'}$ is introduced so that the routing table exchange and thereafter update calculation are done at discrete time. The condition in well formed configuration ensure that the calculi remains consistent in term of self looping of message propagation, path guarantee etc. This calculi presents more realistic picture of distributed networks with routers and therefore is closer to the real implementation.

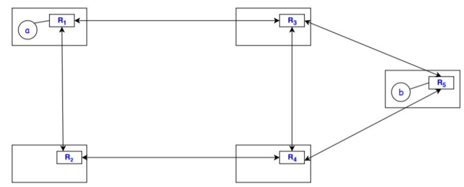

Fig. 1. A simple distributed with routers and nodes

In DR_π^φ, a system is represented by $\langle R^{t_{k'}} \rangle [\![n[P]]\!]$ where a process P is located at node n. The node n is directly connected to the router R at global clock $t = t_{k'}$. The system is accompanied with the router connectivity Γ_c. Hence, the configuration paves the way to reductions. A configuration $\Gamma_c \rhd S$ comprises of router connectivity Γ_c and system S.

In the following sections the paper is organized as follows:

The syntax , structural equivalence and reduction semantics of DR_π^φ are described in Section 2, 3 and 4 respectively. We have explained an example to illustrate the reduction rules more clearly in Section 5. We require certain conditions on well formed configuration for the consistent behavior of the reduction semantics in Section 6. We describe the equivalence between DR_π^φ and D_π in Section 7. The conclusion is in Section 8.

II. SYNTAX

We will use $v, v_1, v_2, u, u_1 \ldots$. to represents values which may be a simple value or a name or a variable. For simplicity in the language we don't use tuples as values. Therefore $u, v, \ldots$.. are singleton names or simple values i.e. integers, boolean etc. We use meta variables $a, b, c, \ldots$ to range over channel names $\complement$ or node names N. In the description of the language $n, m, \ldots$ are used to range over node names N and we use $R, R_1, R_2, \ldots$ to range over set of router names $\Re$ at global time $t_k, t_{k+1}, t_{k+2}, \cdots$. The variables $h, l, \ldots$ range over integers to represent the cost of communication.

Further, we assume that sets of node names, router names and channel names are disjoint from each other. More formally

$$\Re \cap \complement \cap N = \Phi$$

There exists three main syntactic categories in the language that are Nodes , Systems, and Processes. We described the syntax of DR_π^φ in Fig. 1. We have given the descriptions of these syntactic categories in the following sub-sections.

A. System

In Fig. 2, we described a system as $\langle R^{t_{k'}} \rangle [\![M]\!]$ where R being a router at global clock $t_{k'}$ and M is a another syntactic category named as nodes that are directly connected to R. $S \mid T$ represents two parallel systems and $[R] M_{sg}^h (n, m, v@c)$ is a message at router R. This message is used to propagate the value v from one router to another 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. Here the integer h indicates the number of hops(routers), the message has already travelled across the path towards its destination and ε is the identity.

B. Node

In Fig. 2, the nodes are named processes $n[P]$ where n is the name of a node and P is a process term in it. $M \mid N$ describes usual concurrency between nodes M and N at any router. As an example, in a system $\langle R^{t_{k'}} \rangle [\![M \mid N]\!]$ the nodes M and N are running in parallel at router R at global clock $t_{k'}$. 0 is the identity.

C. Process terms

The process terms are very similar to the terms in [1], [5]. These process terms are described in Fig. 2.

III. STRUCTURAL EQUIVALENCE

We introduce a formal relation between the system terms in DR_π^φ called structural equivalence which is represented by the notation $\equiv$ to this relation, they are same computational entity. This is defined in [1], [5]. We describe the definition of structural equivalence is separated for all syntactic categories. Nevertheless, the node equivalence inherits process equivalence and system equivalences inherits by node equivalence. For example, the terms $\langle R^{t_{k'}} \rangle [\![M_1 \mid M_2]\!]$ and $\langle R^{t_{k'}} \rangle [\![M_2 \mid M_1]\!]$, instinctively represent the same systems where the nodes M_1 and M_2 at router R run in parallel at global clock $t_{k'}$ and the order of their composition really does not matter. These are defined in Fig. 2, 3, 4, 5 and 6.

S, T **::=**	**Systems**
$\langle R^{t_{k'}} \rangle [\![M]\!]$	Router at global clock
$S \mid T$	Concurrency
$[R] M_{sg}^h (n, m, v@c)$	Messages
ε	Identity
M, N **::=**	**Nodes**
$n[P]$	Named processes
$M \mid N$	Concurrency
0	Identity
T, U **::=**	**Process Terms**
$c?(x)P$	Input
$m!(v@c)$	Output
if $v_1 = v_2$ then P else Q	Matching
$P \mid Q$	Concurrency
$*P$	Recursive
$stop$	Termination

Fig. 2. Syntax Of DR_π^φ

(SE-COM)	$P \mid Q \equiv Q \mid P$
(SE-ASSOC)	$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$
(SE-ID)	$P \mid id \equiv P$

Fig. 3. Structural Equivalence(Standard) for DR_π^φ

(SE-P-STANDARD)	standard axioms
(SE-P-Recursion)	$* P \equiv P \mid * P$

Fig. 4. Structural Equivalence(Processes) for DR_π^φ

(SE-N-STANDARD)	standard axioms
(SE-N-STOP)	$m[stop] \equiv 0$
(SE-N-INHERITANCE)	$\dfrac{P \equiv Q}{m[P] \equiv m[P]}$

Fig. 5. Structural Equivalence(Nodes) DR_π^φ

(SE-S-STANDARD)	standard axioms
(SE-S-INHERITANCE)	$\dfrac{N \equiv S}{\langle R^{t_{k'}} \rangle [\![N]\!] \equiv \langle R^{t_{k'}} \rangle [\![S]\!]}$

Fig. 6. Structural Equivalence(Systems) for DR_π^φ

IV. REDUCTION SEMANTICS

The reduction semantics are defined on configurations $\Gamma_c \triangleright S$. The configuration reduction step is defined as $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ where the cost of reduction [3], [4] is h and a system S reduces S'. These reduction rules for DR_π^φ are given in Fig. 7 and directly inherited from [8], [10]. The Rule (R-OUT) is for delivery. For example, let us take the configuration $\Gamma_c \triangleright \langle R^{t_{k'}} \rangle [\![n[m!(v@c)$
$|P]|N]\!]$ where a process $m!(v@c)|P$ at source node n at router R at a global clock $t_{k'}$ outputs a value v at channel c which is located at some process at destination node m. This reduction rule generates a propagation message $[R] M_{sg}^0 (n, m, v@c)$ in parallel with the system $\langle R^{t_{k'}} \rangle [\![n[P]|N]\!]$ resulting in a configu-

ration $\Gamma_c \triangleright [R]\, M^0_{sg}\,(n,m,v@c)\,|\langle R^{t_{k'}}\rangle[\![n[P]|N]\!]$. The message with subscript 0 indicates that it has been generated at router R and has not hopped to any other router yet. The term $(n,m,v@c)$ in the message represents the source node name n where a process outputs a value and destination node m where the value v is to be delivered on channel c to a waiting process.

The propagation of the message from one router to another router towards the destination node is done using reduction rule (R-MSG-FWD) in Fig. 7 . Let us consider the reduction rule (R-MSG-FWD). In this reduction rule a configuration of the form $\Gamma_c \triangleright [R_1]M^h_{sg}\,(n,m,v@c)\,|\langle {R_2}^{t_{k'}}\rangle[\![N]\!]|S$ reduced to $\Gamma_c \triangleright [R_2]M^{h+1}_{sg}\,(n,m,v@c)\,|\langle {R_2}^{t_{k'}}\rangle[\![N]\!]|S$. There are two premises the first $(R_1,R_2) \in \Gamma_c$ means that the routers R_1 and R_2 are directly connected or R_2 is a neighbor of R_1. The second one, $\langle {R_1}^{t_{k'}}\rangle(m) = R_2$ means that m belongs to the domain of the routing table at R_1 at global clock $t_{k'}$ and the function $\langle {R_1}^{t_{k'}}\rangle$ returns R_2 as the next hop towards the destination node m.

In the reduction rule (R-COMM) in Fig. 7, a configuration $\Gamma_c \triangleright\ [R]M^h_{sg}\,(n,m,v@c)|\,\langle R^{t_{k'}}\rangle[\![m[c?(x)P|Q]|N]\!]$ does a reduction to $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![m[P\{v/x\}|Q]|N]\!]$. Here note that cost of reduction is h as the message has hopped h routers from source node n to destination node m. However it is not necessary that for every hop the global clock count increases by an interval.

we describe the reduction rule (R-TABLE-UPDATE) in Fig. 7 which uses a new special notation $(\longleftrightarrow)$ to depict the exchange of routing tables at global clock $t = t_{k'}$. This notion $(\longleftrightarrow)$ adds a novelty to this calculus as this will not only exchange the routing table between the connecting routers but also update tables (using distance vector routing methods [13], [14], [15], [16] dynamically with the help of synchronization of the global clock. Here we define reduction semantics for updating table dynamically, may or may not at global clock $t_{k'}$. In this reduction rule a configuration of the form $\langle {R_1}^{t_{k'+1}}\rangle[M]|S$ reduces to $\langle {R'_1}^{t_{k'+1}}\rangle[M]|S$. There are six premises, the first $(R_1,R_2) \in \Gamma_c$ means that the routers R_1 and R_2 are directly connected or R_2 is a neighbor of R_1 at global clock $t = t_{k'}$. The second and third, $\Gamma_c \triangleright \langle {R_1}^{t_{k'}}\rangle[M]|\ S$ and $\langle {R_2}^{t_{k'}}\rangle[N]|T$ are well formed which mean that the well formedness is preserved under reductions. The fourth, $\langle {R_1}^{t_{k'}}\rangle[M]|S \longleftrightarrow \langle {R_2}^{t_{k'}}\rangle[N]|T$ means that the routing table between the connecting routers is exchanged. The fifth and sixth, $\delta\langle {R_1}^{t_{k'+1}}\rangle = \langle {R'_1}^{t_{k'+1}}\rangle$ and $\delta\langle {R_2}^{t_{k'+1}}\rangle = \langle {R'_2}^{t_{k'+1}}\rangle$ mean that routing tables (using distance vector routing methods) are updated dynamically with the help of synchronization of the global clock $t_{k'}$.

The rules (R-MATCH) and (R-MISMATCH) are tests for values. Here the initial cost of these reductions is also zero. The compositional rules are defined in the rule (R-CONTX) in Fig. 7 and are preserved under the static operator $|$. The other reduction rule (R-STRUCT) in Fig. 7 defines well formed configuration reduction upto system structural equivalence.

Now we will demonstrate these rules with the help of an example. This example also shows the exclusive feature of this particular language regarding the novel rule is implemented for routing table updates.

V. Example

In Fig. 1, let us assume that a system S is defined as $S_1|S_2|S_3|S_4|S_5$ where

(R-OUT)
$$\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[m!\,(v@c)\,|P]|N]\!] \rightarrow \Gamma_c \triangleright [R]\,M^0_{sg}\,(n,m,v@c)\,|\langle R^{t_{k'}}\rangle[\![n[P]|N]\!]$$

(R-COMM)
$$\frac{\langle R^{t_{k'}}\rangle(m) = R}{\Gamma_c \triangleright\ [R]M^h_{sg}\,(n,m,v@c)|\,\langle R^{t_{k'}}\rangle[\![m[c?(x)P|Q]|N]\!] \longrightarrow \Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![m[P\{v/x\}|Q]|N]\!]}$$

(R-MSG-FWD)
$$\frac{(R_1,R_2) \in \Gamma_c \qquad \langle {R_1}^{t_{k'}}\rangle(m) = R_2}{\Gamma_c \triangleright [R_1]M^h_{sg}\,(n,m,v@c)\,|\langle {R_2}^{t_{k'}}[\![N]\!]|S \longrightarrow \Gamma_c \triangleright [R_2]M^{h+1}_{sg}\,(n,m,v@c)\,|\langle {R_2}^{t_{k'}}[\![N]\!]|S}$$

(R-MATCH)
$$\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[\text{ if } \ v = v \ \text{ then } P \text{ else } Q]]\!] \longrightarrow \Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[P]]\!]$$

(R-MISMATCH)
$$\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[\text{ if } v_1 \neq v_2 \text{ then } P \text{ else } Q]]\!] \longrightarrow \Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[Q]]\!]$$

(R-TABLE-UPDATE)
$$\frac{\begin{array}{c}(R_1,R_2) \in \Gamma_c,\ t = t_{k'} \\ \Gamma_c \triangleright \langle {R_1}^{t_{k'}}\rangle[M]|\ S \text{ is wff} \\ \Gamma_c \triangleright \langle {R_2}^{t_{k'}}\rangle[N]|T \text{ is wff} \\ \langle {R_1}^{t_{k'}}\rangle[M]|S \longleftrightarrow \langle {R_2}^{t_{k'}}\rangle[N]|T \\ \delta\langle {R_1}^{t_{k'+1}}\rangle = \langle {R'_1}^{t_{k'+1}}\rangle \\ \delta\langle {R_2}^{t_{k'+1}}\rangle = \langle {R'_2}^{t_{k'+1}}\rangle\end{array}}{\begin{array}{c}\Gamma_c \triangleright \langle {R_1}^{t_{k'+1}}\rangle[M]|S \longrightarrow \Gamma_c \triangleright \langle {R'_1}^{t_{k'+1}}\rangle[M]|S \\ \Gamma_c \triangleright \langle {R_2}^{t_{k'+1}}\rangle[N]|T \longrightarrow \Gamma_c \triangleright \langle {R'_2}^{t_{k'+1}}\rangle[N]|T\end{array}}$$

(R-STRUCT)
$$\frac{S \equiv S', \Gamma_c \triangleright S' \longrightarrow \Gamma_c \triangleright R', R' \equiv R}{\Gamma_c \triangleright S \longrightarrow \Gamma_c \triangleright R}$$

(R-CONTX)
$$\frac{\Gamma_c \triangleright S_1 \longrightarrow \Gamma_c \triangleright S'_1}{\begin{array}{c}\Gamma_c \triangleright S_1 \mid S_2 \longrightarrow \Gamma_c \triangleright S'_1 \mid S_2 \\ \Gamma_c \triangleright S_2 \mid S_1 \longrightarrow \Gamma_c \triangleright S_2 \mid S'_1\end{array}}$$

Fig. 7. Reduction Semantics for $\mathrm{DR}^{\varphi}_{\pi}$

$$\begin{array}{c} S_1 \equiv \langle R_1^{t_k}\rangle[\![P|N_1]\!] \\ S_1 \equiv \langle R_2^{t_k}\rangle[\![N_2]\!] \\ S_3 \equiv \langle R_3^{t_k}\rangle[\![N_3]\!] \\ S_4 \equiv \langle R_4^{t_k}\rangle[\![N_4]\!] \\ S_5 \equiv \langle R_5^{t_k}\rangle[\![Q|N_5]\!] \end{array}$$

Where $P \equiv a[b!\langle v@c\rangle]$ and $Q \equiv b[c?(x)R]$
The router connectivity Γ_c is defined as $\{(R_1,R_2),(R_1,R_3),(R_2,R_4),(R_3,R_4),(R_3,R_5),(R_5,R_4)\}$.

The configuration $\Gamma_c \triangleright S_1|S_2|S_3|S_4|S_5$ does a reduction using the rule (R-OUT) where the process $b!\langle v@c\rangle$ at node a generates a message at global clock $t_{k'}$ where $t_{k'} = t_k, t_{k+1}, t_{k+2}, ...$ The configuration reduces to another configuration of the form

$$\Gamma_c \rhd [R_1]\, M^0_{sg}\,(a,b,v@c)\,|\langle R_1^{t_k}\rangle[\![N_1]\!]|S_2|S_3|S_4|S_5$$

In the Fig. 1, R_1 is directly connected to R_2 and R_3. we know that $(R_1,R_2)\in\Gamma_c$ and $(R_1,R_3)\in\Gamma_c$. Similarly we know that $(R_2,R_4)\in\Gamma_c$, $(R_3,R_4)\in\Gamma_c$, $(R_4,R_5)\in\Gamma_c$, $(R_5,R_3)\in\Gamma_c$.

All the routing table will share its routing table with adjacent router. Now by using rule(R-TABLE-UPDATE), we get

$$\begin{gathered}\Gamma_c \rhd \langle R_1^{t_k}\rangle[\![P|N_1]\!] \text{ is wff}\\ \Gamma_c \rhd \langle R_2^{t_k}\rangle[\![N_2]\!] \text{ is wff}\\ \Gamma_c \rhd \langle R_3^{t_k}\rangle[\![N_3]\!] \text{ is wff}\\ \Gamma_c \rhd \langle R_4^{t_k}\rangle[\![N_4]\!] \text{ is wff}\\ \Gamma_c \rhd \langle R_5^{t_k}\rangle[\![Q|N_5]\!] \text{ is wff}\end{gathered}$$

All the systems are well formed which are defined in definition 1. Now all the routing tables shall be exchanged with each other at a global clock t_k.

$$\begin{gathered}\langle R_1^{t_k}\rangle[\![P|N_1]\!] \longleftrightarrow \langle R_2^{t_k}\rangle[\![N_2]\!]\\ \langle R_1^{t_k}\rangle[\![P|N_1]\!] \longleftrightarrow \langle R_3^{t_k}\rangle[\![N_3]\!]\\ \langle R_2^{t_k}\rangle[\![N_2]\!] \longleftrightarrow \langle R_4^{t_k}\rangle[\![N_4]\!]\\ \langle R_3^{t_k}\rangle[\![N_3]\!] \longleftrightarrow \langle R_4^{t_k}\rangle[\![N_4]\!]\\ \langle R_3^{t_k}\rangle[\![N_3]\!] \longleftrightarrow \langle R_5^{t_k}\rangle[\![Q|N_5]\!]\\ \langle R_4^{t_k}\rangle[\![N_4]\!] \longleftrightarrow \langle R_5^{t_k}\rangle[\![Q|N_5]\!]\\ \langle R_5^{t_k}\rangle[\![Q|N_5]\!] \longleftrightarrow \langle R_3^{t_k}\rangle[\![N_3]\!]\end{gathered}$$

Now route checks update for new information then routers will be calculated using Bellman-Ford algorithm and metric is updated, new entries are stored in the routing table. Thus routers will exchange routing information at t_{k+1}.

$$\begin{gathered}\delta\langle {R_1}^{t_{k+1}}\rangle = \langle {R_1}^{t_{k+1}}\rangle\\ \delta\langle {R_2}^{t_{k+1}}\rangle = \langle {R'_2}^{t_{k+1}}\rangle\\ \delta\langle {R_3}^{t_{k+1}}\rangle = \langle {R'_3}^{t_{k+1}}\rangle\\ \delta\langle {R_4}^{t_{k+1}}\rangle = \langle {R_4}^{t_{k+1}}\rangle\\ \delta\langle {R_5}^{t_{k+1}}\rangle = \langle {R'_5}^{t_{k+1}}\rangle\end{gathered}$$

In this way $\langle {R_2}^{t_{k+1}}\rangle$, $\langle {R_3}^{t_{k+1}}\rangle$ and $\langle {R_5}^{t_{k+1}}\rangle$ are updated and new routing tables are $\langle {R'_2}^{t_{k+1}}\rangle$, $\langle {R'_3}^{t_{k+1}}\rangle$ and $\langle {R_5}^{t'_{k+1}}\rangle$ respectively at global clock t_{k+1}.

Now the message hops towards the destination node b, the router table $\langle {R_1}^{t_{k+1}}\rangle$ may return either the adjacent router R_2 or the adjacent router R_3 as next hop on the communication path to node b at router R_5. This may be formally expressed as $\langle {R_1}^{t_{k+1}}\rangle(b) = R_2$ and $\langle {R_1}^{t_{k+1}}\rangle(b) = R_3$.

The communication path is chosen by distance vector approach (shortest path). Suppose the routing table $\langle {R_1}^{t_{k+1}}\rangle$ returns R_3 as the next hop for reaching b. This essentially means that R_3 is on the path towards b which is hosted at router R_5. Formally $\langle {R_1}^{t_{k+1}}\rangle(b) = R_3$ and also we know that $(R_1,R_3)\in\Gamma_c$. Therefore with an application of rule (R-MSG-FWD) the message $[R_1]\,M^0_{sg}(a,b,v@c)$ hops at R_3. So the configuration

$$\begin{gathered}\Gamma_c \rhd [R_1]\, M^0_{sg}\,(a,b,v@c)\,|\langle R_1^{t_{k+1}}\rangle[\![P|N_1]\!]|\langle {R'_2}^{t_{k+1}}\rangle[\![N_2]\!]|\\ \langle {R'_3}^{t_{k+1}}\rangle[\![N_3]\!]|\langle {R_4}^{t_{k+1}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+1}}\rangle[\![Q|N_5]\!]\end{gathered}$$

reduces to

$$\begin{gathered}\Gamma_c \rhd [R_3]\, M^1_{sg}\,(a,b,v@c)\,|\langle R_1^{t_{k+1}}\rangle[\![P|N_1]\!]|\langle {R'_2}^{t_{k+1}}\rangle[\![N_2]\!]|\\ \langle {R'_3}^{t_{k+1}}\rangle[\![N_3]\!]|\langle {R_4}^{t_{k+1}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+1}}\rangle[\![Q|N_5]\!]\end{gathered}$$

Further suppose $\langle {R'_3}^{t_{k+1}}\rangle(b) = R_4$ and the message $[R_3]\,M^1_{sg}(a,b,v@c)$ is propagated to R_4. Since $(R_3,R_4)\in\Gamma_c$. Therefore again using the rule (R-MSG-FWD) the configuration

$$\begin{gathered}\Gamma_c \rhd [R_3]\, M^1_{sg}\,(a,b,v@c)\,|\langle R_1^{t_{k+1}}\rangle[\![P|N_1]\!]|\langle {R'_2}^{t_{k+1}}\rangle[\![N_2]\!]|\\ \langle {R'_3}^{t_{k+1}}\rangle[\![N_3]\!]|\langle {R_4}^{t_{k+1}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+1}}\rangle[\![Q|N_5]\!]\end{gathered}$$

reduces to

$$\begin{gathered}\Gamma_c \rhd [R_4]\, M^2_{sg}\,(a,b,v@c)\,|\langle {R_1}^{t_{k+1}}\rangle[\![P|N_1]\!]|\langle {R'_2}^{t_{k+1}}\rangle[\![N_2]\!]|\\ \langle {R'_3}^{t_{k+1}}\rangle[\![N_3]\!]|\langle {R_4}^{t_{k+1}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+1}}\rangle[\![Q|N_5]\!]\end{gathered}$$

Similarly again all the tables are updated with new entries with an application of rule (R-TABLE-UPDATE) at global clock t_{k+2}. Further suppose $\langle {R'_4}^{t_{k+2}}\rangle(b) = R_5$ and message $[R_4]\,M^2_{sg}(a,b,v@c)$ is propagated to R_5. Since $(R_4,R_5)\in\Gamma_c$. Therefore again using the rule (R-MSG-FWD) the configuration

$$\begin{gathered}\Gamma_c \rhd [R_4]\, M^2_{sg}\,(a,b,v@c)\,|\langle {R'_1}^{t_{k+2}}\rangle[\![P|N_1]\!]|\langle {R''_2}^{t_{k+2}}\rangle[\![N_2]\!]|\\ \langle {R''_3}^{t_{k+2}}\rangle[\![N_3]\!]|\langle {R'_4}^{t_{k+2}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+2}}\rangle[\![Q|N_5]\!]\end{gathered}$$

reduces to

$$\begin{gathered}\Gamma_c \rhd [R_5]\, M^3_{sg}\,(a,b,v@c)\,|\langle {R'_1}^{t_{k+2}}\rangle[\![P|N_1]\!]|\langle {R''_2}^{t_{k+2}}\rangle[\![N_2]\!]|\\ \langle {R''_3}^{t_{k+2}}\rangle[\![N_3]\!]|\langle {R'_4}^{t_{k+2}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+2}}\rangle[\![Q|N_5]\!]\end{gathered}$$

Because $\langle {R'_5}^{t_{k+2}}\rangle(b) = R_5$, the value v is delivered to the waiting process at b using the rule (R-COMM). Therefore the configuration

$$\begin{gathered}\Gamma_c \rhd [R_5]\, M^3_{sg}\,(a,b,v@c)\,|\langle {R'_1}^{t_{k+2}}\rangle[\![P|N_1]\!]|\langle {R''_2}^{t_{k+2}}\rangle[\![N_2]\!]|\\ \langle {R''_3}^{t_{k+2}}\rangle[\![N_3]\!]|\langle {R'_4}^{t_{k+2}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+2}}\rangle[\![b[c?(x)R]|N_5]\!]\end{gathered}$$

reduces to

$$\begin{gathered}\Gamma_c \rhd \langle {R'_1}^{t_{k+2}}\rangle[\![P|N_1]\!]|\langle {R''_2}^{t_{k+2}}\rangle[\![N_2]\!]|\langle {R''_3}^{t_{k+2}}\rangle[\![N_3]\!]|\\ \langle {R'_4}^{t_{k+2}}\rangle[\![N_4]\!]|\langle {R'_5}^{t_{k+2}}\rangle[\![b[R\{v/x\}]|N_5]\!]\end{gathered}$$

Similarly all the tables are updated with new entries by rule (R-TABLE-UPDATE) at every global clock $t_{k'}$ where $t_{k'} = t_k, t_{k+1}, t_{k+2}......$ Thus all the routers in a path of communication between R_1 and R_5 are updated dynamically. This method of routing table update is known as distance vector routing updates.

Previously the path for communication from a to b is via $R_1 \rightsquigarrow R_3 \rightsquigarrow R_5$ where the value propagating message hops two routers before delivering the value at the destination process which means paths are fixed. But now path are changed and new path for communication from a to b via $R_1 \rightsquigarrow R_3 \rightsquigarrow R_4 \rightsquigarrow R_5$. Due to this all the routing tables are updated dynamically. Therefore paths are also changed and this ensures the best optimal path. This is more closer to the real distributed network.

VI. Well Formed Configurations

We define a set of conditions on well formed configurations and prove them in DR^{φ}_{π}. The well formedness is preserved under reductions. The conditions on well formed configurations are explained in definition 1 and DR^{φ}_{π} is ensured by the reduction semantics.

In definition 1, the concept of well formed configurations in DR^{φ}_{π} is inherited from [8] and the reduction rule (6) and (7) are used to illustrate that when (R-COMM) and (R-MSG-FWD) occurs, reduction rule (R-TABLE-UPDATE) is prohibited for given network and vice- versa. These configuration rules will prevent looping and congestion in the network. Hence it will reduce inconsistency in the network.

Definition 1: well formed configuration A configuration is called well formed if it satisfies the following conditions:

1) $\Gamma_c \triangleright \varepsilon$ is a well formed system.
2) If $\Gamma_c \triangleright \langle R^{t_{k'}} \rangle [\![N]\!] \mid S$ is well formed at a global clock if
   a) $\Gamma_c \triangleright S$ is well formed where S contains no message at R.
   b) $\langle R^{t_{k'}} \rangle$ does not occur in S. (Uniqueness of router name R)
   c) $\forall \in \mathrm{fn}(N)$ such that $m \in NN$ where NN is the set of node names, if $\langle R^{t_{k'}} \rangle(m) = R$ then $\forall \ \langle R_1^{t_{k'}} \rangle \in S, \langle R_1^{t_{k'}} \rangle(m) \neq R_1$. (Uniqueness of node name m)
3) If $\Gamma_c \triangleright \langle R^{t_{k'}} \rangle [\![N]\!] \mid S$ is well formed at a global clock $t = t_{k'}, t_{k'+1},$..then $\Gamma_c \triangleright \langle R^{t_{k'+1}} \rangle [\![N]\!] \mid S$ is also well formed.
4) $\Gamma_c \triangleright [R] M^h_{sg}(n, m, v@c) \mid S$ is a well formed if
   a) $\Gamma_c \triangleright S$ is well formed and $S \equiv \langle R^{t_{k'}} \rangle [\![N]\!] \mid S'$ for some S'
   b) There exists a path $P(R', R) = R' \rightsquigarrow R'' \rightsquigarrow$ .....R for some R', R'',.....
   such that $\langle R'^{t_{k'}} \rangle(n) = R'$ and $\langle R''^{t_{k'}} \rangle(m) = R''$..... where $h = \mid \rho(R', R) \mid -1$
5) In any well formed configuration $\Gamma_c \triangleright S$, for every pair of nodes n and m such that $\langle R_i{}^{t_{k'}} \rangle(n) = R_i$ and $\langle R_j{}^{t_{k'}} \rangle(m) = R_j$ at any global clock $t = t_{k'}$ where $(R_i, R_j) \in S$, there exists a unique path $R_i \rightsquigarrow R_j$ such that
   $\langle \ R_i{}^{t_{k'}} \rangle(m) = R', \ \langle \ R'^{t_{k'+1}} \ \rangle(m) = R''$,........$R'^{t_{k'+p}} \rangle(m) = R_j$
6) $\Gamma_c \triangleright S$ is well formed iff
   a) If $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ is using rule (R-COMM) then $\Gamma_c \triangleright S \not\longrightarrow{}^h \Gamma_c \triangleright S'$ will not be used rule (R-TABLE-UPDATE).
   b) If $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ is using rule (R-MSG-FWD) then $\Gamma_c \triangleright S \not\longrightarrow{}^h \Gamma_c \triangleright S'$ will not be used rule (R-TABLE-UPDATE).
7) $\Gamma_c \triangleright S$ is well formed iff
   a) If $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ is using rule (R-TABLE-UPDATE) then
   $\Gamma_c \triangleright S \not\longrightarrow{}^h \Gamma_c \triangleright S'$ will not be used either rule (R-COMM) or rule (R-MSG-FWD) .
   b) If $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ is using rule (R-TABLE-UPDATE) then
   $\Gamma_c \triangleright S \not\longrightarrow{}^h \Gamma_c \triangleright S'$ will not be used rule (R-COMM) and rule (R-MSG-FWD) .

Lemma.1. Suppose $S \equiv T$ then $\Gamma_c \triangleright S$ is well formed iff $\Gamma_c \triangleright T$ is well formed.

Proof. **(OUTLINE)** By induction on definition of $\equiv$.

Theorem 1. *If* $\Gamma_c \triangleright S$ is well formed configuration and $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ then $\Gamma_c \triangleright S'$ is also well formed.

Proof. **(OUTLINE)** By rule induction on inference of $\Gamma_c \triangleright S \longrightarrow^h \Gamma_c \triangleright S'$ then $\Gamma_c \triangleright S'$. It is easy to prove that each inference of $\Gamma_c \triangleright S'$, using the reduction rules in Fig. 7, satisfies all the properties of a well formed configuration.

VII. Equivalence between DR^{φ}_{π} and D_{π}

We proved that whenever a D_π [2] system does a reduction there exists a corresponding well formed configuration in DR^{φ}_{π} which can do a number of reductions such that the residual are equivalent upto structural equivalence after ϕ abstraction of the residual system in DR^{φ}_{π}. Similarly for the converse, we proved that whenever a well formed configuration in DR^{φ}_{π} does a reduction there exists a corresponding D_π system which either does nothing or does a reduction where residuals of both D_π and DR^{φ}_{π} systems are matched upto structural equivalence. Since D_π is a specification for DR^{φ}_{π} therefore we have shown that DR^{φ}_{π} conforms to its specification. Our model is also closer to real distributed networks.

we define a function to abstract away the details of routers and paths from a DR^{φ}_{π} term state theorems about the equivalence of DR^{φ}_{π} with D_π.

Definition 2: We define a function $\phi : LSY \rightarrow HSY$, where LSY and HSY are sets of DR^{φ}_{π} system terms and D_π systems respectively, as follows:

$$\begin{array}{lcl} \phi(\varepsilon) & = & nil \\ \phi(\langle R^{t_{k'}} \rangle [\![N]\!]) & = & N \\ \phi\left([R] M^h_{sg}(n, m, v@c)\right) & = & n[m!\langle v@c \rangle] \\ \phi(S|T) & = & \phi(S) | \phi(T) \end{array}$$

Proposition 1. For any system term L in DR^{φ}_{π} such that $\phi(L) = H$ and $H \equiv H'$ implies that there exists some system term L' in DR^{φ}_{π} such that $\phi(L') = H'$ and $L \equiv L'$.

Proof. We shall prove it by induction on various forms L can take and syntactic analysis of L such that $\phi(L) = H$ and $H \equiv H'$.

1) Let us take a case when a system L, in DR^{φ}_{π}, is of the form
$$L \equiv \langle R_1{}^{t_{k'}} \rangle [\![n[m!\langle v@c \rangle | P] | N]\!]$$
By using ϕ definition, we get
$$\phi(L) = n[m!\langle v@c \rangle | P] | N$$
$$H \equiv n[m!\langle v@c \rangle | P] | N$$
Therefore when we write $\phi(L) = H$ for some term L in DR^{φ}_{π} and a D_π system H , we can rearrange the terms in a D_π systems H', by using various axioms of structural equivalence (SE-COM) and (SE-ID). Since
$$H' \equiv N | n[m!\langle v@c \rangle | P]$$
or
$$H' \equiv n[m!\langle v@c \rangle | P] | N | e$$
Therefore $H \equiv H'$. When a system term L', in DR^{φ}_{π}, is of the form
$$L' \equiv \langle R_1{}^{t_{k'}} \rangle [\![N | n[m!\langle v@c \rangle | P]]\!]$$

Further by definition of ϕ we get,

$$\phi(L') = N|n[m!\langle v@c\rangle|P] \ H' \equiv N|n[m!\langle v@c\rangle|P]$$

By using axiom (SE-COM), we get

$$H' \equiv n[m!\langle v@c\rangle|P]|N \equiv H$$

Now it is clear that, the relation $=$ in the definition ϕ is much stronger than $\equiv$ i.e. ϕ is closed upto $\equiv$. Therefore, by definition of $\equiv$ given in Figure 3 it can be easily verified that $\phi(L') = H'$ and $L \equiv L'$.

2) Let us take another case when a system L, in DR_π^φ, is of the form

$$L \equiv [R_1]M^0_{sg}(n,m,v@c)\,|\langle R_2{}^{t_{k'}}[\![N]\!]|M$$

By using ϕ definition on L, we get

$$\phi(L) = n[m!\langle v@c\rangle|P]|N|M$$
$$H \equiv n[m!\langle v@c\rangle|P]|N|M$$

Therefore when we write $\phi(L) = H$, for some term L in DR_π^φ and D_π system H, we can rearrange the terms in a D_π systems H', by using various axioms of structural equivalence (SE-COM),(SE-ASSOC) and (SE-ID). Since H' is take various forms like,

$$H' \equiv n[m!\langle v@c\rangle|P]|M|N$$
or
$$H' \equiv M|N|n[m!\langle v@c\rangle|P]$$
or
$$H' \equiv n[m!\langle v@c\rangle|P]|N|M|e$$

All the form of H' is structurally equivalent to H, by using various axioms of structural equivalence. Since H'

Now we take system term L' in DR_π^φ, is the form of

$$L' \equiv \langle R_2{}^{t_{k'}}[\![N]\!]|M|[R_1]M^0_{sg}(n,m,v@c)$$

Further by definition of ϕ we get,

$$\phi(L') = N||M|n[m!\langle v@c\rangle|P]$$
$$H' \equiv N||M|n[m!\langle v@c\rangle|P]$$

By using rule (SE-COM), we get

$$H' \equiv n[m!\langle v@c\rangle|P]|N|M \equiv H$$

Therefore $\phi(L') = H'$ and $L \equiv L'$.

Similarly other cases can be proved.

Proposition 2. For any system term L in DR_π^φ $L \equiv L'$ implies $\phi(L) \equiv \phi(L')$.

Proof. This can be proved by induction on the definition of L and $\equiv$ as defined in Figure 2 and . By applying function ϕ above proposition can be derived fairly straightforward.

Lemma.2. In a D_π system H_1 does a reduction $H_1 \longrightarrow H_2$ and $\phi(L_1) = H_1'$ such that $H_1' \equiv H_1$ where L_1 is a system term over a well formed configuration $\Gamma_c \rhd L_1$ in DR_π^φ, then $\Gamma_c \rhd L_1 \longrightarrow \Gamma_c \rhd L_2$ for some h such that $\phi(L_2) = H_2'$ where $H_2' \equiv H_2$.

Proof. We shall prove it by rule induction on the inference of a D_π system reduction $H_1 \longrightarrow H_2$ and syntactic analysis of L_1 such that $\phi(L_1) = H_1'$ where $H_1' \equiv H_1$. There are various possibilities and we we will take each of them as follows:

1) Let us take a case where a D_π system H_1 is the form $l_1[c?(x)P$ $\mid M] \mid l_2[l_1!\langle v@c\rangle \ |N]$. Suppose the D_π system H_1 does a reduction to

$$l_1[P\{v/x\}|M] \mid l_2[N]$$

By using the rule (R-H-COMM) where

$$H_2 \equiv l_1[P\{v/x\}|M] \mid l_2[N]$$

A system term L_1 in DR_π^φ, such that $\phi(L_1) = H_1'$ can take various forms. We shall examine each of them as follows:

a) We take the case where L_1 is structurally equivalent to

$$\langle R_1^{t_{k'}}\rangle[\![l_1[c?(x)P \mid M]]\!] \mid \langle R_2^{t_{k'}}\rangle[\![l_2[l_1!\langle v@c\rangle \ |N]]\!]$$

for some R_1 and R_2. We can clearly see that

$$\phi(L_1) = l_1[c?(x)P \mid M] \mid l_2[l_1!\langle v@c\rangle \ |N]$$

where

$$l_1[c?(x)P \mid M] \mid l_2[l_1!\langle v@c\rangle \ |N] \equiv H_1 \text{ s.t. } H_1' \equiv H_1$$

We know that $\Gamma_c \rhd L_1$ is a well formed system and therefore L_1 does a following reduction using rule (R-OUT) to become

$$[R]M^0_{sg}(l_2,l_1,v@c) \ \mid \langle R_2^{t_{k'}}\rangle[\![l_2[N]]\!] \mid \langle R_1^{t_{k'}}\rangle [\![l_1[c?(x)P \mid M]]\!]$$

We use various standard axioms of structural equivalence rules and by definition of ϕ we know that

$$\phi(L_2) = l_2[l_1!\langle v@c\rangle] \mid l_2[N] \mid l_1[c?(x)P \mid M]$$

where

$$H_2' \equiv l_2[l_1!\langle v@c\rangle] \mid l_2[N] \mid l_1[c?(x)P \mid M]$$

By using axiom (R-H-COM),we get

$$H_2' \equiv l_2[l_1!\langle v@c\rangle] \mid l_1[c?(x)P \mid M]| \ l_2[N]$$

By using axiom (R-H-COMM),we get

$$H_2' \equiv l_1[P\{v/x\}|M] \mid l_2[N] \equiv H_2$$

Further as we know that $\Gamma_c \rhd L_1$ is a well formed system and therefore according to the condition of well formed configuration $R_2 \rightsquigarrow R_1$ where $\langle R_2^{t_{k'}}\rangle(l_2) = R_2$ and $\langle R_1^{t_{k'}}\rangle(l_1) = R_1$. Let us assume that $\langle R_2^{t_{k'}}\rangle(l_2) = R_3$ for some R_3 such that $(R_2,R_3) \in \Gamma_c$. A reduction is done using rule (R-MSG-FWD)

$$[R_2]M^0_{sg}(l_2,l_1,v@c) \ \mid \langle R_2^{t_{k'}}\rangle[\![l_2[N]]\!] \mid \langle R_1^{t_{k'}}\rangle [\![l_1[c?(x)P \mid M]]\!]$$

does a reduction to

$$[R_3]M^1_{sg}(l_2,l_1,v@c) \ \mid \langle R_2^{t_{k'}}\rangle[\![l_2[N]]\!] \mid \langle R_1^{t_{k'}}\rangle [\![l_1[c?(x)P \mid M]]\!]$$

where

$$\phi(L_2) = l_2[l_1!\langle v@c\rangle] \mid l_2[N] \mid l_1[c?(x)P \mid M]$$

and

$$H_2' \equiv l_2[l_1!\langle v@c\rangle] \mid l_2[N] \mid l_1[c?(x)P \mid M]$$

By using axiom (S-MONOID-COM), we get

$$H_2' \equiv l_2[l_1!\langle v@c\rangle] \mid l_1[c?(x)P \mid M] \mid l_2[N]$$

By using axiom (R-H-COMM), we get

$$H_2' \equiv l_1[P\{v/x\}|M] \mid l_2[N] \equiv H_2$$

By using rule (R-COMM), after reduction directly gives the form of H_2.

b) We can take another possibility of the form that a system L_1 in DR_π^φ can take. In a D_π system H_1 is the form $l_1[c?(x)P \mid M] \mid l_2[l_1!\langle v@c\rangle \ |N]$. It is possible that M and N contain several output process terms. Theses output terms will have equivalent messages terms at L_1 which are originated at nodes l_1 and l_2 to carry arbitrary values to some channel at various nodes. Therefore L_1 may contain several messages which may be equivalent to one of the output terms in M or N after ϕ abstraction.

2) We will now consider another possibility when a D_π system H_1 is the form
$$n[\text{ if } v = v \text{ then } P \text{ else } Q]$$
and does a reduction using (R-H-MATCH), we get
$$n[\text{ if } v = v \text{ then } P \text{ else } Q] \rightarrow n[P]$$
where $n[P] \equiv H_2$
In one possibility a system term L_1, in DR^{φ}_{π}, can take a form
$$\langle R^{t_{k'}} \rangle [\![n[\text{ if } v = v \text{ then } P \text{ else } Q]]\!] \equiv L_1$$
for some R such that
$$\phi(L_1) = n[\text{ if } v = v \text{ then } P \text{ else } Q]$$
with an application of rule (R-MATCH) in a well formed configuration $\Gamma_c \triangleright L_1$ can do a reduction to
$$\Gamma_c \triangleright \langle R^{t_{k'}} \rangle [\![n[P]]\!]$$
Here $L_2 \equiv \langle R^{t_{k'}} \rangle [\![n[P]]\!]$. Further with an application of the function ϕ on L_2 we can get
$$\phi(L_1) = n[P] \equiv H'_2$$

3) We will now consider another possibility when a D_π system H_1 is the form
$$n[\text{ if } v_1 = v_2 \text{ then } P \text{ else } Q]$$
and does a reduction using (R-H-MISMATCH), we get
$$n[\text{ if } v = v \text{ then } P \text{ else } Q] \rightarrow n[Q]$$
where $n[Q] \equiv H_2$
In one possibility a system term L_1, in DR^{φ}_{π}, can take a form
$$\langle R^{t_{k'}} \rangle [\![n[\text{ if } v_1 = v_2 \text{ then } P \text{ else } Q]]\!] \equiv L_1$$
for some R such that
$$\phi(L_1) = n[\text{ if } v_1 = v_2 \text{ then } P \text{ else } Q]$$
with an application of rule (R-MISMATCH) in a well formed configuration $\Gamma_c \triangleright L_1$ can do a reduction to
$$\Gamma_c \triangleright \langle R^{t_{k'}} \rangle [\![n[Q]]\!]$$
Here $L_2 \equiv \langle R^{t_{k'}} \rangle [\![n[Q]]\!]$. Further with an application of the function ϕ on L_2 we can get
$$\phi(L_1) = n[Q] \equiv H'_2$$

4) Now we consider the cases of compositional reduction of a D_π system. Let us assume that a D_π system H_1 is of the form $P_1 \mid P_2$. An application of the rule (R-H-CONTX) reduces H_1 to $P'_1 \mid P_2$. Let us assume that a system term in DR^{φ}_{π}, is of the form $L_1 \mid L_2$ such that $\phi(L_1) = P_1$ and $\phi(L_2) = P_2$. We also assume that a configurations $\Gamma_c \triangleright L_1 \mid L_2$ and $\Gamma_c \triangleright L_1$ are well formed configurations. By induction we can say that $\Gamma_c \triangleright L_1 \longrightarrow \Gamma_c \triangleright L'_1$ such that $\phi(L'_1) = P''_1$ for some h, $P''_1 \equiv P'_1$. Therefore using the rule (R-CONTX), we can conclude that $\Gamma_c \triangleright L_1 \mid L_2 \longrightarrow \Gamma_c \triangleright L'_1 \mid L_2$. We know that $\phi(L'_1 \mid L_2) = \phi(L'_1) \mid \phi(L_2)$. Since $\phi(L_2) = P_2$ and $\phi(L'_1) = P''_1$ therefore $\phi(L'_1 \mid L_2) = P''_1 \mid P_2$. Further we already know that $P''_1 \equiv P'_1$ therefore from the axioms of structural equivalence , we can conclude that $P'_1 \mid P_2 \equiv H_2$.
Same as H_1 is the form of $P_2 \mid P_1$. We can proved similarly.

5) Let us now consider the last case when a D_π system H_1 does a reduction to H_2 using the rule (R-H-STRUCT) because $H'_1 \longrightarrow H'_2$ where $H_1 \equiv H'_1$ and $H_2 \equiv H'_2$. Let us assume that for a system L_1 in DR^{φ}_{π}, $\phi(L_1) = H_1$. We also assume that $\Gamma_c \triangleright L_1$ is a well formed configuration. Since $H_1 \equiv H'_1$ therefore using proposition 1. we can say that there exists a L'_1 such that $\phi(L'_1) = H'_1$ and $L \equiv L'$. Further using lemma.1. We know that $\Gamma_c \triangleright L'_1$ is a well formed. Now by induction we can say $H'_1 \longrightarrow H'_2$ implies $\Gamma_c \triangleright L'_1 \longrightarrow \Gamma_c \triangleright L''_1$ for some L''_1 and h such that $\phi(L''_1) = H''_2$ for some H''_2 such that $H''_2 \equiv H'_2$. We already know that $L_1 \equiv L'_1$ therefore with an application of rule (R-STRUCT). We can say that $\Gamma_c \triangleright L_1 \xrightarrow{*} \Gamma_c \triangleright L''_1$. We know that $\phi(L''_1) = H''_2$ and since $H''_2 \equiv H'_2$, $H'_2 \equiv H_2$ therefore $H''_2 \equiv H_2$

Lemma.3. In DR^{φ}_{π}, if a well formed configuration $\Gamma_c \triangleright L_1$ does a reduction $\Gamma_c \triangleright L_1 \longrightarrow \Gamma_c \triangleright L_2$ and $\phi(L_1) = H_1$ where H_1 is a D_π system then either there exists a D_π system H_2 such that $H_1 \longrightarrow H_2$ and $\phi(L_2) \equiv H_2$ or $\phi(L_2) \equiv H_1$.

Proof. By induction on the inference of reduction $\Gamma_c \triangleright L_1 \longrightarrow \Gamma_c \triangleright L_1$ of well formed configurations in DR^{φ}_{π} and syntactic analysis of $\phi(L_1) = H_1$. There are various possibilities and we we will take each of them as follows:

1) Let us take a case when a system L_1, in DR^{φ}_{π}, is of the form $\langle R^{t_{k'}} \rangle [\![n[m!\langle v@c \rangle |P]|N]\!]$. A well formed configuration $\Gamma_c \triangleright L_1$ does a reduction to
$$\Gamma_c \triangleright [R]\, M^0_{sg}\,(n, m, v@c)\, \langle R^{t_{k'}} \rangle [\![n[P]|N]\!]$$
using rule (R-OUT) in fig. 7, Let a D_π system H_1 is of the form $n[m!\langle v@c \rangle |P]|N$ and by definition of ϕ we know that
$$\phi(L_1) = \phi(\langle R^{t_{k'}} \rangle [\![n[m!\langle v@c \rangle |P]|N]\!]) = n[m!\langle v@c \rangle |P]|N$$
Further by definition of ϕ we know that
$$\phi(L_2) = \phi([R]\, M^0_{sg}\,(n, m, v@c)\, \langle R^{t_{k'}} \rangle [\![n[P]|N]\!]) = n[m!\langle v@c \rangle |P]|N$$
now by an application of axiom (S-H-MERGE), We can conclude that
$$n[m!\langle v@c \rangle |P]|N \equiv H_1$$

2) In another case we consider that a system term L_1, in DR^{φ}_{π}, is of the form $[R_1]M^h_{sg}\,(n, m, v@c)\,|\langle R_2{}^{t_{k'}} [\![N]\!] |S$. We consider a case when using the rule (R-MSG-FWD), the well formed configuration $\Gamma_c \triangleright L_1$ does a reduction to $\Gamma_c \triangleright [R_2]M^{h+1}_{sg}$ $(n, m, v@c)\,|\langle R_2{}^{t_{k'}} [\![N]\!] |S$. Let a D_π system H_1 be of the form $n\,[m!\langle v@c \rangle]\,|N|S$ where $\phi(L_1) = H_1$. Clearly
$$\phi(L_1) = \phi([R_1]M^h_{sg}\,(n, m, v@c)\,|\langle R_2{}^{t_{k'}} [\![N]\!] |S)$$
$$\phi(L_1) = n\,[m!\langle v@c \rangle]\,|N|S \equiv H_1$$
Further by definition of ϕ we know that
$$\phi(L_2) = \phi([R_2]M^{h+1}_{sg}\,(n, m, v@c)\,|\langle R_2{}^{t_{k'}} [\![N]\!] |S)$$
$$\phi(L_2) = n\,[m!\langle v@c \rangle]\,|N|S \equiv H_1$$

3) Now let us take a case when a system L_1, in DR^{φ}_{π}, is of the form

$[R]M^{h}_{sg}(n,m,v@c)|\langle R^{t_{k'}}\rangle[\![m[c?(x)P|Q]|N]\!]$. The well formed configuration $\Gamma_c \triangleright L_1$ does a reduction using the rule (R-COMM) to another well formed configuration $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![m[P\{v/x\}|Q]|N]\!]$. Let us assume that $L_2 = R^{t_{k'}}\rangle[\![m[P\{v/x\}|Q]|N]\!]$. Clearly

$$\phi(L_1) = n[m!\langle v@c\rangle]\,|m[P\{v/x\}|Q]|N$$

By using the rule (R-H-COMM), The D_π system $n[m!\langle v@c\rangle]\,|m[P\{v/x\}|Q]|N$ can reduce to $m[P\{v/x\}|Q]|n[\varepsilon]|N$ which is structurally equivalent to $m[P\{v/x\}|Q]|N \equiv H_1$. We know that $\phi(L_2) = m[P\{v/x\}|Q]|N \equiv H_1$

4) Now we take another case where a system term L_1, in DR^{φ}_{π}, is of the form either $\Gamma_c \triangleright \langle {R_1}^{t_{k'+1}}\rangle[M]|S$ or $\Gamma_c \triangleright \langle {R_2}^{t_{k'+1}}\rangle[N]|T$. We can take one form $\Gamma_c \triangleright \langle {R_1}^{t_{k'+1}}\rangle[M]|S$. The well formed configuration $\Gamma_c \triangleright L_1$ reduces using the rule (R-TABLE-UPDATE) to another well formed configuration $\Gamma_c \triangleright \langle {R'_1}^{t_{k'+1}}\rangle[M]|S$ and by definition of ϕ we know that

$$\phi(L_1) = M|S \equiv H_1$$
$$\phi(L_2) = M|S \equiv H_1$$

5) Now we take another case where a system term L_1, in DR^{φ}_{π}, is of the form $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[\text{ if } v = v \text{ then } P \text{ else } Q]]\!]$. The well formed configuration $\Gamma_c \triangleright L_1$ reduces using the rule (R-MATCH) to another well formed configuration $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[P]]\!]$ and by definition of ϕ we know that

$$\phi(L_1) = n[\text{ if } v = v \text{ then } P \text{ else } Q]$$
$$\phi(L_1) = n[P] \equiv H_1 \text{ where value is true then result is } P$$
$$\phi(L_2) = n[P] \equiv H_1$$

6) Now we take another case where a system term L_1, in DR^{φ}_{π}, is of the form $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[\text{ if } v_1 \neq v_2 \text{ then } P \text{ else } Q]]\!]$. The well formed configuration $\Gamma_c \triangleright L_1$ reduces using the rule (R-MISMATCH) to another well formed configuration $\Gamma_c \triangleright \langle R^{t_{k'}}\rangle[\![n[Q]]\!]$ and by definition of ϕ we know that

$$\phi(L_1) = n[\text{ if } v_1 \neq v_2 \text{ then } P \text{ else } Q] = n[Q] \equiv H_1$$
$$\phi(L_1) = n[Q] \equiv H_1$$

7) Now let us take compositional cases. First suppose a system in in DR^{φ}_{π}, is of the form $L_1 \mid L_2$ and a D_π system is of the form $H_1 \mid H_2$ where $\phi(L_1) = H_1$ and $\phi(L_2) = H_2$. By definition of the ϕ we can clearly see that $\phi(L_1 \mid L_2) = H_1 \mid H_2$. Now take the case when the well formed configuration in DR^{φ}_{π}, $\Gamma_c \triangleright L_1 \mid L_2$ does a reduction using the rule (R-CONTX) to another well formed configuration $\Gamma_c \triangleright L'_1 \mid L_2$, the well formed configuration $\Gamma_c \triangleright L_1$ does a reduction to $\Gamma_c \triangleright L_1 \longrightarrow \Gamma_c \triangleright L'_1$ for some h. By induction we know that
   a) either $H_1 \longrightarrow H'_1$ such that $\phi(L'_1) = H'_1$
   b) or $\phi(L'_1) = H_1$

   From the reduction rule reduction rule (R-H-CONTX), the D_π system $H_1 \mid H_2$ can reduce to $H'_1 \mid H_2$. We already know that either $\phi(L'_1) = H'_1$ or $\phi(L'_1) = H_1$, therefore by definition of ϕ we know that, either $\phi(L'_1 \mid L_2) \equiv H'_1 \mid H_2$ or $\phi(L'_1 \mid L_2) \equiv H_1 \mid H_2$. The other case in rule (R-CONTX) is exactly similar.

8) let us take second compositional case where a well formed configuration in DR^{φ}_{π}, $\Gamma_c \triangleright L_1$ does a reduction to $\Gamma_c \triangleright L_2$ using the rule (R-STRUCT)because

$$\Gamma_c \triangleright L'_1 \longrightarrow \Gamma_c \triangleright L'_2$$

for some h where $L_1 \equiv L'_1$ and $L_2 \equiv L'_2$. Let us assume that a D_π system H_1 is such that $\phi(L_1) = H_1$. As $L_1 \equiv L'_1$ therefore from proposition 2, we know that $\phi(L_1) \equiv \phi(L'_1)$. Now by induction we know that $\Gamma_c \triangleright H_1 \longrightarrow \Gamma_c \triangleright H_2$ for some h the D_π system term H_2 such that either $\phi(L'_2) \equiv H_2$ or $\phi(L'_2) \equiv H_1$. Since it is known that $L_2 \equiv L'_2$ and using proposition 2 we know that $\phi(L'_2) \equiv \phi(L_2)$ therefore either $\phi(L_2) \equiv H_2$ or $\phi(L_2) \equiv H_1$.

Theorem 2. In DR^{φ}_{π}, if a well formed configuration $\Gamma_c \triangleright L_1$ does a reduction $\Gamma_c \triangleright L_1 \longrightarrow \Gamma_c \triangleright L_2$ and $\phi(L_1) = H_1$ where H_1 is a D_π system if and only if either there exists a D_π system H_2 such that $H_1 \longrightarrow H_2$ and $\phi(L_2) \equiv H_2$ or $\phi(L_2) \equiv H_1$.

*Proof.***(OUTLINE)** By using lemma.2 and lemma.3.

VIII. Conclusion

We described the syntax and reduction semantics for the calculus DR^{φ}_{π} that gives a realistic model of distributed network with incorporation of dynamic updation in routing table. We have explained an example to demonstrate reduction rules and also demonstrate that how routing table is updated across the network by using distance vector routing updates. This equivalence has been proved with the well known distributed π-calculus, D_π after abstracting away the unnecessary details from DR^{φ}_{π}. Now we have proved that both DR^{φ}_{π} and D_π systems are reduction equivalent after abstracting away the details of routers and paths from DR^{φ}_{π}.

This calculus implemented routing table updates via distance vector routing methods and included the exclusive feature of this particular calculus with a novel notation ($\longleftrightarrow$) to depict the exchange of routing tables at global time $t = t_{k'}$. Thus all the routing tables are updated dynamically due to this paths are also changed and ensure the best optimal path. This is more close to the actual real distributed network.

In DR^{φ}_{π}, the δ function used in the calculus is abstract function which does not allow a real value in the network. Also the calculus does not support dynamic node creation which can be a possible future work for further research. In this paper, we have shown that specification coincides its implementation in our next work will justify the calculus using bisimulation based proof technique.

References

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

[2] M. 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*, vol. 48, pp. 23–32., May 2008.

[3] M. Gaur, "A routing calculus: Towards formalising the cost of computation in a distributed computer network," PhD, Informatics, University of Sussex, U.K., December 2008.

[4] M. Gaur and M. Hennessy, ". counting the cost in the picalculus (extended abstract)." *Electronic Notes in Theoretical Computer Science (ENTCS)*, pp. 229:117–129, 2009.

[5] M. Hennessy., *A distributed Pi-Calculus.* Cambridge University Press, 2007.

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

[7] J. Parrow., *An Introduction to the π -Calculus*, Dep. Teleinformatics,Royal Institue of Technology,Stockholm.

[8] M. Gaur, S. J. Gay, and I. Mackie, "A routing calculus with flooding updates," in *Proceedings of the 11th International Conference on Distributed Computing and Internet Technology - Volume 8956*, ser. ICDCIT 2015. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 181–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-14977-6_12

[9] R. Milner, *Communicating and mobile systems: The Pi Calculus*. Cambridge: Cambridge University Press, 1999.

[10] M. Gaur., "A routing caculus towards formalising the cost of computation in a distributed computer netwok," Ph.D. dissertation, University of Sussex, 2008.

[11] U. Montanari and M. Sammartino, "Network-conscious π-calculus - A model of pastry," *Electr. Notes Theor. Comput. Sci.*, vol. 312, pp. 3–17, 2015. [Online]. Available: https://doi.org/10.1016/j.entcs.2015.04.002

[12] R. D. Nicola, G. L. Ferrari, and R. Pugliese, "Klaim: a kernel language for agents interaction and mobility," *IEEE Transactions on Software Engineering*, 1998.

[13] L. Tang and Q. Liu, "A survey on distance vector routing protocols," *CoRR*, vol. abs/1111.1514, 2011. [Online]. Available: http://arxiv.org/abs/1111.1514

[14] J. J. Garcia-Luna-Aceves, "A unified approach to loop-free routing using distance vectors or link states," *SIGCOMM Comput. Commun. Rev.*, vol. 19, no. 4, pp. 212–223, Aug. 1989. [Online]. Available: http://doi.acm.org/10.1145/75247.75268

[15] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 92–93, Jun. 2002. [Online]. Available: http://doi.acm.org/10.1145/581291.581305

[16] I. Vukotic and S. Scepanovic, "Description of distance vector routing protocols using mathematical means," 01 2013.

[17] A. S. Tanenbaum., *Computer Networks.*, 4th ed. Upper Saddle River,New Jersey,: Pearson Education, Inc.,, 2003.

[18] C. Cheng, R. Riley, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves, "A loop-free extended bellman-ford routing protocol without bouncing effect," in *Symposium Proceedings on Communications Architectures &Amp; Protocols*, ser. SIGCOMM '89. New York, NY, USA: ACM, 1989, pp. 224–236. [Online]. Available: http://doi.acm.org/10.1145/75246.75269

[19] D. Walden, "The bellman-ford algorithm and "distributed bellman-ford," 01 2008.

[20] K. R. Hutson, T. L. Schlosser, and D. R. Shier, "On the distributed bellman-ford algorithm and the looping problem," vol. 19, pp. 542–551, 11 2007.