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 $D^\omega$ 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 $D^\pi$ (distributed $\pi$-calculus) after abstracting away the unnecessary details from our calculus which in fact is one of the implementations of $D^\pi$. We nomenclate our calculus with routing table updates as $D^\omega$. We have explained an example to illustrate the reduction rules in our semantics rule. Further to maintain the consistency in the calculi a clock $t^\pi$ 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.

In $D^\omega$, a system is represented by $\langle R^\omega \rangle \vec{[n[P]]}$ where a process $P$ is located at node $n$. The node $n$ is directly connected to the router $R$ at global clock $t=t^\omega$. The system is accompanied with the router connectivity $\Gamma_c$. Hence, the configuration paves the way to reductions. A configuration $\Gamma_c \triangleright S$ comprises of router connectivity $\Gamma_c$ and system $S$.

In the following sections the paper is organized as follows:

The syntax, structural equivalence and reduction semantics of $D^\omega$ 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 $D^\omega$ and $D^\pi$ in Section 7. The conclusion is in Section 8.

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 $D^\omega$ 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 $D^\pi$ (distributed $\pi$-calculus) after abstracting away the unnecessary details from our calculus which in fact is one of the implementations of $D^\pi$. We nomenclate our calculus with routing table updates as $D^\omega$. We have explained an example to illustrate the reduction rules in our semantics rule. Further to maintain the consistency in the calculi a clock $t^\pi$ 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.

In $D^\omega$, a system is represented by $\langle R^\omega \rangle \vec{[n[P]]}$ where a process $P$ is located at node $n$. The node $n$ is directly connected to the router $R$ at global clock $t=t^\omega$. The system is accompanied with the router connectivity $\Gamma_c$. Hence, the configuration paves the way to reductions. A configuration $\Gamma_c \triangleright S$ comprises of router connectivity $\Gamma_c$ and system $S$.

In the following sections the paper is organized as follows:

The syntax, structural equivalence and reduction semantics of $D^\omega$ 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 $D^\omega$ and $D^\pi$ 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 \( \mathcal{C} \) 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 \( \mathcal{R} \) at global time \( t_k, t_{k+1}, t_{k+2}, \ldots \). 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

\[
\mathcal{R} \cap \mathcal{C} \cap N = \Phi
\]

There exists three main syntactic categories in the language that are Nodes, Systems, and Processes. We described the syntax of \( \text{DR}_\mathcal{R} \) 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 \( \langle R^{v'} \rangle \mathcal{M} \) where \( R \) being a router at global clock \( t_v \) and \( M \) is another syntactic category named as nodes that are directly connected to \( R \). \( S \mid T \) represents two parallel systems and \( [R]M^{h_0}_{\mathcal{R}}(n, m, v@c) \) 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 \( e \) 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 \( \langle R^{v'} \rangle \mathcal{M} \mid \mathcal{N} \) the nodes \( M \) and \( N \) are running in parallel at router \( R \) at global clock \( t_v \). 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 \( \text{DR}_\mathcal{R} \) 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 \( \langle R^{v'} \rangle \mathcal{M}_1 \mid \mathcal{M}_2 \) and \( \langle R^{v'} \rangle \mathcal{M}_2 \mid \mathcal{M}_1 \), instinctively represent the same systems where the nodes \( M_1 \) and \( M_2 \) at router \( R \) run in parallel at global clock \( t_v \) and the order of their composition really does not matter. These are defined in Fig. 2, 3, 4, 5 and 6.

IV. REDUCTION SEMANTICS

The reduction semantics are defined on configurations \( \Gamma_\mathcal{C} \triangleright S \). The configuration reduction step is defined as \( \Gamma_\mathcal{C} \triangleright S \quad \Gamma_\mathcal{C} \triangleright S' \) where the cost of reduction [3], [4] is \( h \) and a system \( S \) reduces \( S' \). These reduction rules for \( \text{DR}_\mathcal{R} \) 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 \( \Gamma_\mathcal{C} \triangleright \langle R^{v'} \rangle \mathcal{N} \) where a process \( m!(v@e) \mid P \) at source node \( n \) at router \( R \) at a global clock \( t_v \) outputs a value \( v \) at channel \( c \) which is located at some process at destination node \( m \). This reduction rule generates a propagation message \( [R]M^{h_0}_{\mathcal{R}}(n, m, v@c) \) in parallel with the system \( \langle R^{v'} \rangle[n[P] \mid N] \) resulting in a configu-
\[ \Gamma_c \triangleright [R]M_{sg}^h(n,m,v@c) \langle [R^\epsilon] \rangle \langle n|P \rangle |N]. \]

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 \( \Gamma_c \triangleright [R]M_{sg}^h(n,m,v@c) \langle [R^\epsilon] \rangle \langle n|P \rangle |N] \) reduced to \( \Gamma_c \triangleright [R_1]M_{sg}^{h+1}(n,m,v@c) \langle [R_2^\epsilon] \rangle |N]|S \). There are two premises of the first \((R_1R_2) \in \Gamma_c\) means that the routers \( R_1 \) and \( R_2 \) are directly connected or \( R_2 \) is a neighbor of \( R_1 \). The second one, \( \langle R_1^\epsilon \rangle (m) = R_2 \) means that \( m \) belongs to the domain of the routing table at \( R_1 \) at global clock \( t_\epsilon \) and the function \( \langle R_1^\epsilon \rangle \) returns \( R_2 \) as the next hop towards the destination node \( m \).

In the reduction rule (R-COMM) in Fig. 7, a configuration \( \Gamma_c \triangleright [R]M_{sg}^h(n,m,v@c) \langle [R^\epsilon] \rangle \langle m|c?x)P \rangle Q \rangle |N] \) does a reduction to \( \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|P[v/x]\rangle Q \rangle |N]\). Here note that cost of reduction is \( h \) as the message has 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 \( \langle \rightarrow \rangle \) to depict the exchange of routing tables at clock \( t = t_\epsilon \). This notion \( \langle \rightarrow \rangle \) 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 \( t_\epsilon \). In this reduction rule a configuration of the form \( \langle R_1^\epsilon \rangle |M]\langle S \) reduces to \( \langle R_1^{\epsilon+1} \rangle |M]\langle S \). 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 clock \( t = t_\epsilon \). The second and third, \( \Gamma_c \triangleright \langle R_1^\epsilon \rangle |M]\langle S \rangle \triangleright \langle R_2^\epsilon \rangle |N]\langle T \rangle \) are well formed which mean that the well formedness is preserved under reductions. The fourth, \( \langle R_1^\epsilon \rangle |M]\langle S \rangle \leftrightarrow \langle R_2^\epsilon \rangle |N]\langle T \rangle \) means that the routing table between the connecting routers is exchanged. The fifth and sixth, \( \delta(R_1^{\epsilon+1}) = \langle R_1^{\epsilon+1} \rangle \) and \( \delta(R_2^{\epsilon+1}) = \langle R_2^{\epsilon+1} \rangle \) mean that routing tables (using distance vector routing methods) are updated dynamically with the help of synchronization of the global clock \( t_\epsilon \).

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-CONTEX) in Fig. 7 and are preserved under the static operator \( \cdot \). The other reduction rule (R-STRUCT) in Fig. 7 defines well formed configuration reduction upon 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)
\[ \Gamma_c \triangleright [R^\epsilon \langle n|v@c \rangle |P \rangle |N] \rightarrow \Gamma_c \triangleright [R]M_{sg}^h(n,m,v@c) \langle [R^\epsilon] \rangle \langle n|P \rangle |N] \]

(R-COMM)
\[ \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|v@c \rangle |P \rangle |N] \rightarrow \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|P[v/x]\rangle Q \rangle |N] \]

(R-MSG-FWD)
\[ (R_1,R_2) \in \Gamma_c \triangleright \langle R_1^\epsilon \rangle (m) = R_2 \]

(R-REPLACE)
\[ \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|c?x)P \rangle Q \rangle |N] \]

(R-MISMATCH)
\[ \Gamma_c \triangleright \langle R^\epsilon \rangle \langle n|v_1 \neq v_2 \rangle P \rangle |N] \rightarrow \Gamma_c \triangleright \langle R^\epsilon \rangle \langle n|Q \rangle \]

(R-TABLE-UPDATE)
\[ \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|P[v/x]\rangle Q \rangle |N] \rightarrow \Gamma_c \triangleright \langle R^\epsilon \rangle \langle m|P \rangle |N] \]

(S\rangle = \langle \rightarrow \rangle \] where \( m \) belongs to the domain of the routing table at \( R_1 \) at global clock \( t_\epsilon \) and the function \( \langle R_1^\epsilon \rangle \) returns \( R_2 \) as the next hop towards the destination node \( m \).

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

\[ S_1 \equiv \langle R_1^1 \rangle |P|N_1 \]
\[ S_3 \equiv \langle R_3^2 \rangle |N_2 \]
\[ S_4 \equiv \langle R_4^3 \rangle |N_3 \]
\[ S_5 \equiv \langle R_5^4 \rangle |Q|N_3 \]

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 \( b!v@c \) at node \( a \) generates a message at global clock \( t_\epsilon \) where \( t_\epsilon = t_1k_1 + t_2k_2 + \ldots \). The configuration reduces to another configuration of the form

\[ \ldots \]

www.ijacsa.thesai.org 558 | P a g e
In the Fig. 1, $R_1$ is directly connected to $R_2$ and $R_3$, we know that $(R_1, R_2) \in \Gamma$, and $(R_1, R_3) \in \Gamma$. Similarly we know that $(R_2, R_4) \in \Gamma_c$, $(R_3, R_4) \in \Gamma_c$, $(R_1, R_5) \in \Gamma_c$, $(R_5, R_3) \in \Gamma_c$.

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

$\Gamma_c \vdash [R_1] M^1_{R_1} (a, b, v @ c) \land (R_1^{k+1}) \land (P) \land (N_1) \land (R_1^{k+1}) \land (N_2) \land (R_1^{k+1}) \land (Q) \land (N_5) \land (\Phi) \land (N_3)$

Further suppose $(R_4^{k+1}) (b) = R_4$ and the message $[R_1] M^1_{R_1} (a, b, v @ c)$ is propagated to $R_4$. Since $(R_1, R_4) \in \Gamma_c$. Therefore again using the rule (R-MSG-FWD) the configuration

$\Gamma_c \vdash [R_4] M^2_{R_4} (a, b, v @ c) \land (R_1^{k+1}) \land (P) \land (N_1) \land (R_1^{k+1}) \land (N_2) \land (R_1^{k+1}) \land (Q) \land (N_5) \land (\Phi) \land (N_3)$

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 $(R_1^{k+1}) (b) = R_2$ and message $[R_1] M^2_{R_1} (a, b, v @ c)$ is propagated to $R_1$. Since $(R_1, R_1) \in \Gamma_c$. Therefore again using the rule (R-MSG-FWD) the configuration

$\Gamma_c \vdash [R_2] M^2_{R_2} (a, b, v @ c) \land (R_1^{k+1}) \land (P) \land (N_1) \land (R_1^{k+1}) \land (N_2) \land (R_1^{k+1}) \land (Q) \land (N_5) \land (\Phi) \land (N_3)$

Therefore again using the rule (R-COMM). Therefore the configuration

$\Gamma_c \vdash [R_5] M^3_{R_5} (a, b, v @ c) \land (R_1^{k+1}) \land (P) \land (N_1) \land (R_1^{k+1}) \land (N_2) \land (R_1^{k+1}) \land (Q) \land (N_5) \land (\Phi) \land (N_3)$

Because $(R_1^{k+1}) (b) = R_3$, the value $v$ is delivered to the waiting process at $b$ using the rule (R-COMM). Therefore the configuration

$\Gamma_c \vdash [R_3] M^3_{R_3} (a, b, v @ c) \land (R_1^{k+1}) \land (P) \land (N_1) \land (R_1^{k+1}) \land (N_2) \land (R_1^{k+1}) \land (Q) \land (N_5) \land (\Phi) \land (N_3)$

Similarly all the tables are updated with new entries by rule (R-TABLE-UPDATE) at every global clock $t_i \forall i$ where $t_i' = t_{i+k+1}, t_{k+2}, \ldots$. Thus all the routers in a path of communication between $R_1$ and $R_3$ 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 $R_1 \rightarrow R_3 \rightarrow R_5$, 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 $R_1 \rightarrow R_3 \rightarrow R_4 \rightarrow R_5$. 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. The well formedness is preserved under reductions. The conditions on well formed configurations are explained in definition 1 and DR_ϕ is ensured by the reduction semantics.

In definition 1, the concept of well formed configurations in DR_ϕ 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. \( \Gamma_c \triangleright e \) is a well formed system.
2. If \( \Gamma_c \triangleright (\{R\}^k)[N] \) | \( S \) is well formed at a global clock if
   a. \( \Gamma_c \triangleright S \) is well formed where \( S \) contains no message at \( R \).
   b. \( \langle R^k \rangle \) does not occur in \( S \). (Uniqueness of router name \( R \))
   c. \( \forall \in \text{fin}(N) \) such that \( m \in NN \) where \( NN \) is the set of node names, if \( \langle R^k \rangle (m) = R \) then \( \forall (R^k) \in S, (R^k)(m) \neq R_1 \). (Uniqueness of node name \( m \))
3. If \( \Gamma_c \triangleright (\{R\}^k)[N] \) | \( S \) is well formed at a local clock \( t \equiv t_1, t_2, \ldots, t_n \) then \( \Gamma_c \triangleright (\{R\}^k)[N] \) | \( S \) is also well formed.
4. \( \Gamma_c \triangleright [R]M_0^0(n,m,v@c) \) | \( S \) is a well formed if
   a. \( \Gamma_c \triangleright S \) is well formed and \( S \equiv \langle R^k \rangle [N] \) | \( S' \) for some \( S' \)
   b. There exists a path \( P(R',R) = R' \rightarrow R'' \rightarrow \ldots \rightarrow R \) for some \( R', R'', \ldots \) such that \( \langle R^k \rangle(m) = R' \) and \( \langle R^k \rangle(m) = R'' \)...
   c. \( h = \rho(R',R') \) \( \triangleright -1 \)
5. In any well formed configuration \( \Gamma_c \triangleright S \), for every pair of nodes \( n \) and \( m \) such that \( \langle R^k \rangle(m) = R_i \) and \( \langle R^k \rangle(n) = R_j \) at any global clock \( t \equiv t_1 \) where \( R_i, R_j \in S \), there exists a unique path \( R_i \rightarrow R_j \) such that \( \langle R^k \rangle(m) = R_i \) \( \langle R^k \rangle(n) = R_j \), and \( R_i \rightarrow R_j \)
6. \( \Gamma_c \triangleright S \) is well formed iff
   a. If \( \Gamma_c \triangleright S \rightarrow^{b} \Gamma_c \triangleright S' \) is using rule (R-COMM) then \( \Gamma_c \triangleright S \rightarrow^{h} \Gamma_c \triangleright S' \) will not be used rule (R-TABLE-UPDATE).
   b. If \( \Gamma_c \triangleright S \rightarrow^{a} \Gamma_c \triangleright S' \) is using rule (R-MSG-FWD) then \( \Gamma_c \triangleright S \rightarrow^{h} \Gamma_c \triangleright S' \) will not be used rule (R-TABLE-UPDATE).
7. \( \Gamma_c \triangleright S \) is well formed iff
   a. If \( \Gamma_c \triangleright S \rightarrow^{b} \Gamma_c \triangleright S' \) is using rule (R-COMM) then \( \Gamma_c \triangleright S \rightarrow^{h} \Gamma_c \triangleright S' \) will not be used either rule (R-COMM) or rule (R-MSG-FWD).
   b. If \( \Gamma_c \triangleright S \rightarrow^{a} \Gamma_c \triangleright S' \) is using rule (R-COMM) then \( \Gamma_c \triangleright S \rightarrow^{h} \Gamma_c \triangleright S' \) 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 \( \Gamma_c \triangleright S \rightarrow^{b} \Gamma_c \triangleright S' \) then \( \Gamma_c \triangleright S' \) is also well formed.

**Proof (OUTLINE)** By rule induction on inference of \( \Gamma_c \triangleright S \rightarrow^{b} \Gamma_c \triangleright S' \) 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 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 which can do a number of reductions such that the residual are equivalent upto structural equivalence after \( \phi \) abstraction of the residual system in DR_ϕ. Similarly for the converse, we proved that whenever a well formed configuration in DR_ϕ 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 are matched upto structural equivalence. Since D_π is a specification for DR_ϕ therefore we have shown that DR_ϕ 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 term state theorems about the equivalence of DR_ϕ with D_π.

**Definition 2:** We define a function \( \phi : LSY \rightarrow HSY \), where LSY and HSY are sets of DR_ϕ system terms and D_π systems respectively, as follows:

\[
\begin{align*}
\phi(e) & = \text{nil} \\
\phi((R^k)[N]) & = N \\
\phi([R[M_0^0(n,m,v@c)]]) & = [n[m!(v@c)]] \\
\phi(S(T)) & = \phi(S) \phi(T) \\
\end{align*}
\]

**Proposition 1.** For any system term \( L \) in DR_ϕ such that \( \phi(L) = H \) and \( H \equiv H' \) implies that there exists some system term \( L' \) in DR_ϕ such that \( \phi(L') = H' \).

**Proof.** We shall prove it by induction on various forms \( L \) and symbolic 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, is of the form

\[
L \equiv [R_1^k][n[m!(v@c)]P]\]

By using \( \phi \) definition, we get

\[
\begin{align*}
\phi(L) & = n[m!(v@c)]P \\
H & = n[m!(v@c)]P \\
\end{align*}
\]

Therefore when we write \( \phi(L) = H \) for some term \( L \) in DR_ϕ 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

\[
H' \equiv N[n[m!(v@c)]P]
\]

or

\[
H' \equiv n[m!(v@c)]P]
\]

Therefore \( H \equiv H' \). When a system term \( L' \), in DR_ϕ, is of the form

\[
L' \equiv [R_1^k][N[n[m!(v@c)]P]]
\]
Further by definition of $\phi$ we get,

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

By using axiom (SE-COM), we get

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

Now it is clear that the relation $\equiv$ in the definition $\phi$ is much stronger than $\equiv$ i.e. $\phi$ is closed up to $\equiv$. 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 $\text{DR}_\Gamma^\phi$ is of the form

$$L \equiv [R_1]\ M^\phi_{\text{sg}} (n,m,v@c) \langle R_2 ^{\phi'} \langle N\rangle | M$$

By using $\phi$ definition on $L$, we get

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

or

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

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

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

or

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

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

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

Now we take system term $L'$ in $\text{DR}_\Gamma^\phi$, is the form of

$$L' \equiv \langle R_2 ^{\phi'} \langle N\rangle | M[R_1]\ M^\phi_{\text{sg}} (n,m,v@c)$$

Further by definition of $\phi$ we get,

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

By using rule (SE-COM), we get

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

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

Similarly other cases can be proved.

**Proposition 2.** For any system term $L$ in $\text{DR}_\Gamma^\phi$ $L \equiv L'$ 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 $\phi$ above proposition can be derived fairly straightforward.

**Lemma 2.** In a $\text{D}_\Gamma$ system $H_1$ does a reduction $H_1 \rightarrow H_2$ and $\phi(L_1) = H_1' \equiv H_1'$ such that $H_2' \equiv H_1'$ where $L_1$ is a system term over a well formed configuration $\Gamma_c \triangleright L_1$ in $\text{DR}_\Gamma^\phi$, then $\Gamma_c \triangleright H_1 \rightarrow \Gamma_c \triangleright H_2$ 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 $\text{D}_\Gamma$ system reduction $H_1 \rightarrow H_2$ 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 will take each of them as follows: 

1) Let us take a case where a $\text{D}_\Gamma$ system $H_1$ is the form

$$l_1[c?(x)\langle P]\ | M] \ l_2[l_1[v@c]\ | N].$$

Suppose the $\text{D}_\Gamma$ system $H_1$ does a reduction to

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

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

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

A system term $L_1$ in $\text{DR}_\Gamma^\phi$, 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^L\phi \langle l_1[c?(x)\langle P]\ | M]\ \langle R^K\phi \langle l_2[l_1[v@c]\ | N]\$$

for some $R_1$ and $R_2$. We can clearly see that $\phi(L_1) = l_1[c?(x)\langle P]\ | M] \ l_2[l_1[v@c]\ | N]$ where

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

Now we have $\Gamma_c \triangleright L_1$ is a well formed system and therefore $L_1$ does a following reduction using rule (R-OUT) to become

$$\langle R^M\phi \langle l_2[l_1[v@c]\ | R^K\phi \langle l_2[l_1[v@c]\ | N]\$$

We use various standard axioms of structural equivalence rules and by definition of $\phi$ we know that

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

where

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

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

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

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

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

Further as we know that $\Gamma_c \triangleright L_1$ is a well formed system and therefore according to the condition of well formed configuration $R_2 \rightarrow R_1$ where $\langle R^L\phi \langle l_2[l_1[v@c]\ | R^K\phi \langle l_2[l_1[v@c]\ | N]\$, then $\Gamma_c \triangleright H_1 \rightarrow \Gamma_c \triangleright H_2$ for some $h$ such that $\phi(L_2) = H_2'$ where $H_2' \equiv H_2$.

By using the rule (R-H-COMM), we get

$$\langle R^M\phi \langle l_2[l_1[v@c]\ | R^K\phi \langle l_2[l_1[v@c]\ | N]\$$

where

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

and

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

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

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

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

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

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 $\text{DR}_\Gamma^\phi$ can take. In a $\text{D}_\Gamma$ system $H_1$ is the form

$$l_1[c?(x)\langle P]\ | M] \ l_2[l_1[v@c]\ | N].$$

It is possible that $M$ and $N$ contain several output

$$\langle R^L\phi \langle l_1[c?(x)\langle P]\ | l_2[l_1[v@c]\ | N]\$$

www.ijacsa.thesai.org
process terms. Theses output terms will have equivalent message 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 \( \phi \) abstraction.

2) We will now consider another possibility when a \( D_\pi \) system \( H_1 \) is the form

\[ n \mid \text{if } v = v \text{ then } P \text{ else } Q \]

and does a reduction using (R-H-MATCH), we get

\[ n \mid \text{if } v = v \text{ then } P \text{ else } Q \rightarrow n[P] \]

where \( n[P] \equiv H_2 \)

In one possibility a system term \( L_1 \), in \( DR_\pi^\phi \), can take a form

\[ \langle R' \rangle[n] \mid \text{if } v = v \text{ then } P \text{ else } Q \]  

\[ = L_1 \]

for some \( R \) such that

\[ \phi(L_1) = n \mid \text{if } v = v \text{ then } P \text{ else } Q \]

with an application of rule (R-MATCH) in a well formed configuration \( \Gamma \rightarrow L_1 \) can do a reduction to

\[ \Gamma \rightarrow \langle R' \rangle[n[P]] \]

Here \( L_2 \equiv \langle R' \rangle[n[P]] \). Further with an application of the function \( \phi \) on \( L_2 \) we can get

\[ \phi(L_1) = n[P] \equiv H_2' \]

3) We will now consider another possibility when a \( D_\pi \) system \( H_1 \) is the form

\[ n \mid \text{if } v_1 = v_2 \text{ then } P \text{ else } Q \]

and does a reduction using (R-H-MISMATCH), we get

\[ n \mid \text{if } v = v \text{ then } P \text{ else } Q \rightarrow n[Q] \]

where \( n[Q] \equiv H_2 \)

In one possibility a system term \( L_1 \), in \( DR_\pi^\phi \), can take a form

\[ \langle R' \rangle[n] \mid \text{if } v_1 = v_2 \text{ then } P \text{ else } Q \]  

\[ = L_1 \]

for some \( R \) such that

\[ \phi(L_1) = n \mid \text{if } v_1 = v_2 \text{ then } P \text{ else } Q \]

with an application of rule (R-MISMATCH) in a well formed configuration \( \Gamma \rightarrow L_1 \) can do a reduction to

\[ \Gamma \rightarrow \langle R' \rangle[n[Q]] \]

Here \( L_2 \equiv \langle R' \rangle[n[Q]] \). Further with an application of the function \( \phi \) 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 \( D_\pi \) system. Let us assume that a \( D_\pi \) 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 \( DR_\pi^\phi \), 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 \( \Gamma \rightarrow L_1 \mid L_2 \) and \( \Gamma \rightarrow L_1 \) are well formed configurations. By induction we can say that \( \Gamma \rightarrow L_1 \mid L_2 \rightarrow \Gamma \rightarrow L_1' \mid L_2 \), where \( \Gamma \rightarrow L_1' \mid L_2 \) is a well formed configuration. Therefore \( H_1 \equiv H_1' \) using proposition 1. We can say that there exists a \( L_1' \). Similarly, using lemma.1. Further using lemma.1, we can say that there exists a \( L_1' \). Similarly, using lemma.1. Now by an application of \( R-\text{H-CONTX} \) implies \( \Gamma \rightarrow L_1' \mid L_2' \rightarrow \Gamma \rightarrow L_1'' \mid L_2'' \) for some \( L_1'' \) and \( L_2'' \) 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-CONTX).

5) Let us now consider the last case when a \( D_\pi \) system \( H_1 \) does a reduction to \( H_2 \) using the rule (R-H-STRUCT) because \( H_1 \rightarrow H_2 \) where \( H_1 \equiv H_1' \) and \( H_2 \equiv H_2' \). Let us assume that for a system \( L_1 \) in \( DR_\pi^\phi \), \( \phi(L_1) = H_1 \). We also assume that \( \Gamma \rightarrow L_1 \) 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 \( \Gamma \rightarrow L_1' \) is a well formed. Now by induction we can say \( H_1' \rightarrow H_2' \) implies \( \Gamma \rightarrow L_1' \rightarrow \Gamma \rightarrow L_1'' \) 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 \( \Gamma \rightarrow L_1 \rightarrow \Gamma \rightarrow L_1' \). We know that \( \phi(L_1') = H_2'' \) and since \( H_2'' \equiv H_2' \), \( H_2' = H_2 \) therefore \( H_2'' \equiv H_2 \).

Lemma.3. In \( DR_\pi^\phi \), if a well formed configuration \( \Gamma \rightarrow L_1 \) does a reduction \( \Gamma \rightarrow L_1 \rightarrow \Gamma \rightarrow L_2 \) and \( \phi(L_1) = H_1 \) where \( H_1 \) is a \( D_\pi \) system then either there exists a \( D_\pi \) system \( H_2 \) such that \( H_1 \rightarrow H_2 \) and \( \phi(L_2) \equiv H_2 \) or \( \phi(L_2) = H_1 \).

Proof. By induction on the inference of reduction \( \Gamma \rightarrow L_1 \rightarrow \Gamma \rightarrow L_2 \) of well formed configurations in \( DR_\pi^\phi \) and syntactic analysis of \( \phi(L_1) = H_1 \). There are various possibilities and we will take each of them as follows:

1) Let us take a case when a system \( L_1 \), in \( DR_\pi^\phi \), is of the form \( \langle R' \rangle[n[m(v@c)]P\|N] \). A well formed configuration \( \Gamma \rightarrow L_1 \) does a reduction to

\[ \Gamma \rightarrow \langle R' \rangle[n[m \mid (v@c)]P\|N] \]

using rule (R-OUT) in fig. 7. Let a \( D_\pi \) system \( H_1 \) of the form \( n[m!(v@c)]P\|N \) and by definition of \( \phi \) we know that

\[ \phi(L_1) = \phi(\langle R' \rangle[n[m(v@c)]P\|N]) = n[m!(v@c)]P\|N \]

Further by definition of \( \phi \) we know that

\[ \phi(L_2) = \phi(\langle R \rangle[M_{sg}^0(n,m,v@c)]R'\|N\|P\|N]) = n[m!(v@c)]P\|N \]

now by an application of axiom (S-H-MERGE), we can conclude that

\[ n[m!(v@c)]P\|N \equiv H_1 \]

2) In another case we consider that a system term \( L_1 \), in \( DR_\pi^\phi \), is of the form \( \langle R_1 \rangle[M_{sg}^0(n,m,v@c)]\langle R_2 \rangle[V']\|N\|S \). We consider a case when using the rule (R-MSG-FWD), the well formed configuration \( \Gamma \rightarrow L_1 \) does a reduction to \( \Gamma \rightarrow \langle R_1 \rangle[M_{sg}^0(n,m,v@c)]\langle R_2 \rangle[V'\|N]\|S \). Let a \( D_\pi \) system \( H_1 \) be of the form \( n[m!(v@c)]\|N\|S \) where \( \phi(L_1) = H_1 \). Clearly

\[ \phi(L_1) = \phi(\langle R_1 \rangle[M_{sg}^0(n,m,v@c)]\langle R_2 \rangle[V'\|N]\|S]) = n[m!(v@c)]\|N\|S \equiv H_1 \]

Further by definition of \( \phi \) we know that

\[ \phi(L_2) = \phi(\langle R_2 \rangle[M_{sg}^0(n,m,v@c)]\langle R_2 \rangle[V']\|N]) = n[m!(v@c)]\|N\|S \equiv H_1 \]

3) Now let us take a case when a system \( L_1 \), in \( DR_\pi^\phi \), is of the form
Now we take another case where a system term $L_v$ does a reduction using the rule (R-COMM) to another well formed configuration $\Gamma_v \triangleright L_1$. Let us assume that $L_2 = \langle R_v \rangle [\Pi][P_{v/x}]Q[N]$. Clearly

$$\phi(L_1) = n[m!(v@e)]m[P_{v/x}]Q[N]$$

By using the rule (R-H-COMM), the $\text{D}_\pi$ system $m[n!(v@e)]m[P_{v/x}]Q[N]$ can reduce to $m[P_{v/x}]Q[N]$ which is structurally equivalent to $m[P_{v/x}]Q[N]$. We know that $\phi(L_2) = \phi(L_1)$. Now we take another case where a system term $L_1$, in $\text{D}_\pi^\rho$, is of the form either $\Gamma_v \triangleright (\langle R_v \rangle[n][v = v \text{ then } P \text{ else } Q])$. The well formed configuration $\Gamma_v \triangleright L_1$ reduces using the rule (R-MATCH) to another well formed configuration $\Gamma_v \triangleright (\langle R_v \rangle[n][P])$ and by definition of $\phi$ we know that

$$\phi(L_1) = n[v = v \text{ then } P \text{ else } Q] \equiv H_1$$

Now we take another case where a system term $L_1$, in $\text{DR}_\pi^\rho$, is of the form either $\Gamma_v \triangleright (\langle R_v \rangle[n][v_1 \neq v_2 \text{ then } P \text{ else } Q])$. The well formed configuration $\Gamma_v \triangleright L_1$ reduces using the rule (R-MISMATCH) to another well formed configuration $\Gamma_v \triangleright (\langle R_v \rangle[n][Q])$ and by definition of $\phi$ we know that

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

Now let us take compositional cases. First suppose a system in in $\text{DR}_\pi^\rho$, is of the form $L_1 \vdash L_2$ and a $\text{D}_\pi$ system is of the form $H_1 \vdash H_2$ where $\phi(L_1) = H_1$ and $\phi(L_2) = H_2$. By definition of $\phi$ we can clearly see that $\phi(L_1) \vdash L_2 = H_1 \vdash H_2$. Now take the case when the well formed configuration in $\text{DR}_\pi^\rho$, $\Gamma_v \triangleright L_1 \vdash L_2$ does a reduction using the rule (R-MISMATCH) to another well formed configuration $\Gamma_v \triangleright L_1 \vdash L_2$ and the well formed configuration $\Gamma_v \triangleright L_1$ does a reduction to $\Gamma_v \triangleright L_2$ for some $h$. From induction we know that

a) either $H_1 \vdash H_1'$ such that $\phi(L_1') = H_1'$

b) or $\phi(L_1') = H_1'$

From the reduction rule reduction rule (R-H-CONTX), the $\text{D}_\pi$ system $H_1 \vdash H_2$ can reduce to $H_1' \vdash H_2$. We already know that either $\phi(L_1') = H_1'$ or $\phi(L_1') = H_1$, therefore by definition of $\phi$ we know that, either $\phi(L_1') \vdash L_2 \equiv H_1' \vdash H_2$ or $\phi(L_1') \vdash L_2 \equiv H_1 \vdash H_2$. The other case in rule (R-CONTX) is exactly similar.

8) let us take second compositional case where a well formed configuration in $\text{DR}_\pi^\rho$, $\Gamma_v \triangleright L_1$ does a reduction to $\Gamma_v \triangleright L_2$ using the rule (R-STRUCT). Because

$$\Gamma_v \triangleright L_1' \rightarrow \Gamma_v \triangleright L_2'$$

for some $h$ where $L_1' \equiv L_1'$ and $L_2' \equiv L_2'$. Let us assume that a $\text{D}_\pi$ 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 $\Gamma_v \triangleright H_1 \rightarrow \Gamma_v \triangleright H_2$ for some $h$ the $\text{D}_\pi$ system term $H_2$ such that either $\phi(L_1') \equiv H_2$ or $\phi(L_1') \equiv H_1$. Since it is known that $L_2 \equiv L_2'$ and using proposition 2 we know that $\phi(L_2) \equiv \phi(L_1)$ therefore either $\phi(L_2) \equiv H_2$ or $\phi(L_2) \equiv H_1$. Therefore, $\phi(L_2) \equiv \phi(L_1')$ or $\phi(L_2) \equiv H_1$. Thus by using lemma 2 and lemma 3.

VIII. Conclusion

We described the syntax and reduction semantics for the calculus $\text{DR}_\pi^\rho$ that gives a realistic model of distributed network with incorporation of dynamic update 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 $\pi$-calculus, $\text{D}_\pi$ after abstracting away the unnecessary details from $\text{DR}_2$. Now we have proved that both $\text{DR}_\pi^\rho$ and $\text{D}_\pi$ systems are reduction equivalent after abstracting away the details of routers and paths from $\text{DR}_2$.

This calculus implemented routing table updates via distance vector routing methods and included the exclusive feature of this particular calculus with a novel notation $(\rightarrow)$ to depict the exchange of routing tables at global time $t = t_c$. 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 $\text{DR}_\pi^\rho$, the $\delta$ 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 justifiy the calculus using bisimulation based proof technique.

REFERENCES


