$DR_\pi^\omega F$: A Fault Tolerant Distributed Routing Calculi

Manish Gaur
Institute of Engg. & Tech.
Sitapur Road,Lucknow
manish.gaur@ietlucknow.ac.in

Rama Kant
Anand Engg. college
Keetham, Agra
baghel_kant@rediffmail.com

ABSTRACT

We design a new language $DR_\pi^\omega F$ in which we extend the language DR_π^ω to handle node failure during communication. We have extended the reduction semantics of DR_π^ω. For handling the node failure we describe two tables at router: One is routing table contains name mapping for forward travel of messages and other is a database of all active nodes at any router. Further we discuss the new node creation and failed node activation during communication. We show how the entries of routing tables are updated using backward learning updates. We explain the usefulness and working of our reduction semantics by taking one example.

General Terms
Pi Calculus, Formal Methods, Routing
Keywords
Routing calculi,node failure,Fault Tolerance

1 INTRODUCTION

Nowadays interactive systems permeate our everyday life such as online shopping, online banking, social networking sites, ATM machines etc. We are so advanced technologically to meet these requirements but design principles and techniques for assuring their correct behaviour are at much more primitive stage. There are many approaches used to meet these requirements. One of them is developing a **Formal Calculi** in which the fundamental concepts of underlying interactive systems can be described and studied. There are many earlier developments describing the abstract behaviour of systems such as **CCS** [14, 13, 12] and **Pi-Calculus**[14, 13, 12].

Asynchronous Pi-Calculus [8] is one such calculi describing the evolution and behaviour of a asynchronous communication system. In this language, values can be exchanged between concurrent processes via communication channels. Communication channels can be used to model resources and the syntax allows them to be declared as private, for the exclusive shared use of specific processes. The names of these channels/resources can also be transmitted between processes. The other developments are Distributed Pi-Calculus [8] and routing calculi ,DR_π^ω[5] and DR_π [6]. Many other various developments are discussed in [7, 17, 9, 16, 15, 11].

The routing calculi ,DR_π^ω[5] and DR_π [6], are developed with the intention of modeling a distributed network to demonstrate the cost of communication between the communicating processes. The cost of communication is the number of hops (router) a value propagating message crosses before delivering it to the destination in a network of routers. In fact the value propagating messages used in these models closely resemble the IP packet in TCP/IP [1, 2, 10, 18, 19] model of networks. However ,the names (addresses) of source and destination nodes and data in messages are used unlike real IP packets where lots of other information is contained in it. The crucial role of routers in determining the quality of communication services in a distributed network is demonstrated in both the calculi.The difference between DR_π[6] and DR_π^ω [5] is that DR_π^ω [5] uses backward learning update of routing tables while DR_π[6] uses flooding for update routing tables.

In this short paper we show how the reduction semantics which are developed for $DR_\pi^\omega F$ works under distributed systems with node failure. we describe a language which consists of a network of routers where routers connectivity is fixed . The processes reside in a located site called nodes. Nodes are directly connected

S,T::=	**Systems**
$\langle$ R$\rangle$ $[\![$ M $]\!]$	Router
S | T	Concurrency
[R]M^k_{sg}(n,m,v@c)	Messages
(new d)M	New Name
ε	Identity

M,N::=	**Nodes**
n[T]	Named Processes
M | N	Concurrency
(new d)M	New Name
0	Identity

T,U::=	**Process Terms**
c?(x)T	Input
m!$\langle$ v@c $\rangle$	Output
if u = v then T else U	Matching
(new b)T	Channel Name creation
newnode m with P in Q	New Node creation
T | U	Concurrency
*T	Repetition
stop	Identity

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

to some specific router and any two processes at nodes can communicate via the routers. Nodes may fail during communication between processes. A considerable amount of work has been done along the line in [4, 3].

We discuss syntax,structural equivalence and reduction semantics in section 2. We also discuss one example to show the application of reduction semantics in section 3.

(S-SCOPE-EXTRUSION)	(new d)(P| Q)	$\equiv$	P | (new d) Q, if d $\notin$ fn(P)
(S-MONOID-COM)	P | Q	$\equiv$	Q| P
(S-MONOID-ASSOC)	(P | Q) | R	$\equiv$	P | (Q | R)
(S-MONOID-ID)	P | id	$\equiv$	P
(S-NEW-FLIP)	(new c) (new d) P	$\equiv$	(new d) (new c) P
(S-NEW)	(new d) P	$\equiv$	P , if d $\notin$ fn(P)
(S-NEW-ID)	(new d) id	$\equiv$	id

Figure 2: Structural Equivalence (Standard) for $DR^{\omega}_{\pi}F$

(S-P-STANDARD)	standard axioms		
(S-P-UNWIND)	*P	$\equiv$	P| *P

Figure 3: Structural Equivalence (Processes) for $DR^{\omega}_{\pi}F$

2 $DR^{\omega}_{\pi}F$: A FAULT TOLERANT DISTRIBUTED ROUTING CALCULI

2.1 Syntax

The syntax of the language $DR^{\omega}_{\pi}F$ is given in Figure 1. We will use v, v_1, v_2, u, u_1, u_2 . . . to describe values which may be a name or a variable or a simple value. For simplicity in the language we don't use tuples as values. Therefore v, u, . . . are singleton names or simple values i.e. integers, boolean etc. We use meta variables a, b, c, . . . to range over sets of channel names CN or node names NN. In the description of the language n,m, . . . are used to range over set of node names NN and we use R,R_1,R_2, . . . to range set of router names RN. The variables k, l, . . . range over set of integers to represent the cost of communication. Further, we assume that sets of channel names, node names and router names are disjoint from each other. More formally

$$RN \cap NN \cap CN = \phi$$

There are three syntactic categories in the language. These are Systems, Nodes and Processes.

2.2 Structural Equivalence

The concepts of structural equivalence in $DR^{\omega}_{\pi}F$, where two terms are said to be structurally equivalent if they are same computational entity. For example , the terms $\langle$ R $\rangle$ $[\![$ M | N $]\!]$ and $\langle$ R $\rangle$ $[\![$ N | M $]\!]$ intuitively represent the same systems where the nodes M and N at router R are running in parallel; the order of their

(S-P-STANDARD)	standard axioms		
(S-N-STOP)	m[stop]	$\equiv$	0
(S-N-INHERITANCE)	$\dfrac{P \equiv Q}{m[P] \equiv m[Q]}$		
(S-N-EXTR)	m[(new d)P]	$\equiv$	(new d)m[P], if d$\neq$ m

Figure 4: Structural Equivalence (Nodes) for DR$^{\omega}_{\pi}$F

(S-SY-STANDARD)	standard axioms		
(S-SY-INHERITANCE)	$\dfrac{N \equiv S}{\langle R\rangle \llbracket N \rrbracket \equiv \langle R\rangle \llbracket S \rrbracket}$		
(S-SY-EXTR)	(new d)$\langle$ R$\rangle$ $\llbracket$ N $\rrbracket$	$\equiv$	$\langle$ R $\rangle$ $\llbracket$ (new d)M$\rrbracket$,if d$\neq$ R

Figure 5: Structural Equivalence (Systems) for DR$^{\omega}_{\pi}$F

composition really does not matter. Similarly in the systems $\langle$ R $\rangle$ $\llbracket$ m[P $\mid$ Q]$\rrbracket$ and $\langle$ R $\rangle$ $\llbracket$ m[Q $\mid$ P]$\rrbracket$ the order of composition of processes running at node m at router R is not important and they represent the same system as well. However, it is very important to note that in DR$^{\omega}_{\pi}$F , α -conversion is only applied to the channel names of processes and not to the node and system names. The names of routers and nodes are fixed and disjoint. They can not be renamed or altered at any stage.

The axioms which are specific to process, nodes and system equivalence are stated in Figure 2, Figure 4 and Figure 5 respectively. There are certain axioms of structural equivalence which are standard and applicable to all syntactic categories so they are separated and described in Figure 2. In Figure 2, the terms used as P, Q, R can be a process term, node or system. Only terms in the same syntactic category can have structural equivalence.

Therefore for example in Figure 2 when using the axiom (S-MONOID-COM) we say that P $\mid$ Q $\equiv$ Q $\mid$ P then we mean that this axiom can be applied when P and Q both are any one of system terms, node terms or process terms. The term used as id refers to the respective identity elements in each syntactic category. The identity for system, node and process is ε , 0 and stop respectively.

2.3 Reduction Semantics

To handle node failure routers maintain two tables: a routing table, $\langle R^t \rangle$, consists of a mapping for each node name to some adjacent router which is on the

(R-OUT-NF)

$$\frac{\langle R^a\rangle[n]\downarrow}{\Gamma_c \rhd \langle R\rangle\llbracket n[m!\langle v@c\rangle \mid P] \mid N\rrbracket \longrightarrow \Gamma_c \rhd [R]M^0_{sg}(n,m,v@c) \mid \langle R\rangle\llbracket n[P] \mid N\rrbracket}$$

(R-MSG-FWD-NF-I)

$$\frac{(R_1,R_2)\in\Gamma_c \quad \langle R^t_1\rangle(m)=R_2 \quad \langle R^t_2\rangle(v)\uparrow,\ \ if\ \ v\in NodeName}{\Gamma_c \rhd [R_1]M^k_{sg}(n,m,v@c) \mid \langle R_2\rangle\llbracket N\rrbracket \mid S \longrightarrow \Gamma_c \rhd [R_2]M^{k+1}_{sg}(n,m,v@c) \mid \langle R^t_2\{m\to R_1\}^+\rangle\llbracket N\rrbracket \mid S}$$

(R-MSG-FWD-NF-II)

$$\frac{(R_1,R_2)\in\Gamma_c \quad \langle R^t_1\rangle(m)=R_2 \quad \langle R^t_2\rangle(v)\downarrow,\ \ if\ \ v\in NodeName}{\Gamma_c \rhd [R_1]M^k_{sg}(n,m,v@c) \mid \langle R_2\rangle\llbracket N\rrbracket \mid S \longrightarrow \Gamma_c \rhd [R_2]M^{k+1}_{sg}(n,m,v@c) \mid \langle R_2\rangle\llbracket N\rrbracket \mid S}$$

(R-IN-NF-I)

$$\frac{\langle R^a\rangle[m]\downarrow \quad \langle R^t\rangle(m)=R}{\Gamma_c \rhd [R]M^k_{sg}(n,m,v@c) \mid \langle R\rangle\llbracket m[c?(x)P \mid Q] \mid N\rrbracket \longrightarrow^k \Gamma_c \rhd \langle R\rangle\llbracket m[P\{^v/x\} \mid Q] \mid N\rrbracket}$$

(R-IN-NF-II)

$$\frac{\langle R^a\rangle[m]\uparrow \quad \langle R^t\rangle(m)=R}{\Gamma_c \rhd [R]M^k_{sg}(n,m,v@c) \mid \langle R\rangle\llbracket m[P] \mid N\rrbracket \mid S \longrightarrow \Gamma_c \rhd \langle R^t\{m\to R\}^-, R^a\{m\}^-\rangle\llbracket N\rrbracket \mid S}$$

(R-NEWNODE-CREATION-ACTIVATION)

Γ_c $\rhd$ $\langle$ R$\rangle$ $\llbracket$ n[newnode m with P in Q]$\rrbracket$ $\longrightarrow$
Γ_c $\rhd$ (new m) ($\langle R^t\{m\to R\}^+, R^a\{m\}^+\rangle$ $\llbracket$ n[Q]$\mid$ m[P]$\rrbracket$)

(R-MATCH)

Γ_c $\rhd$ $\langle$ R$\rangle$ $\llbracket$ n[if v=v then P else Q]$\rrbracket$ $\longrightarrow$ Γ_c $\rhd$ $\langle$ R$\rangle$ $\llbracket$ n[P]$\rrbracket$

(R-MISMATCH)

Γ_c $\rhd$ $\langle$ R$\rangle$ $\llbracket$ n[if v_1=v_2 then P else Q]$\rrbracket$ $\longrightarrow$
Γ_c $\rhd$ $\langle$ R$\rangle$ $\llbracket$ n[Q]$\rrbracket$, $v_1 \neq v_2$

(R-STRUCT)

$$\frac{S\equiv S', \Gamma_c \rhd S' \longrightarrow^k \Gamma_c \rhd R', R'\equiv R}{\Gamma_c \rhd S \longrightarrow^k \Gamma_c \rhd R}$$

(R-CONTX)

$$\frac{\Gamma_c \rhd S_1 \longrightarrow^k \Gamma_c \rhd S'_1}{\begin{array}{l}\Gamma_c \rhd S_1 \mid S_2 \longrightarrow^k \Gamma_c \rhd S'_1 \mid S_2\\ \Gamma_c \rhd S_2 \mid S_1 \longrightarrow^k \Gamma_c \rhd S_2 \mid S'_1\\ \Gamma_c \rhd (new\ \ k)S_1 \longrightarrow^k \Gamma_c \rhd (new\ \ k)S'_1\end{array}}$$

Figure 6: Reduction Semantics for DR$^{\omega}_{\pi}$F

path of the destination node and a active node table , $\langle R^a \rangle$, maintains the set of active node at router R in a system. In DR$^{\omega}_{\pi}$F we will represent the combination of routing table, $\langle R^t \rangle$ and active node table,$\langle R^a \rangle$ by router table, $\langle$ R$\rangle$ as needed. The status of a node is detected automatically at home router i.e. node is failed or get activated.

On detection of the node failure, the router handles the node failure with the help of these tables. Suppose a node m fails at router R then two actions are performed at router R: First node m is deleted from active node table $\langle R^a \rangle$; this is formally represented as $\langle R^a\{m\}^- \rangle$ and second entry $\langle R^t \rangle$ (m)=R is deleted from $\langle R^t \rangle$; this is formally represented as $\langle R^t\{m \rightarrow R\}^- \rangle$. In another words the node m is deleted from router R which is the result of these two atomic actions.

Upon creation of new nodes (or activation of failed node) , we describe the method of updating the routing table $\langle R^t \rangle$ and active node table $\langle R^a \rangle$ of home router. The routing table of other router which either use this new node name (or recently activated node name) in the propagating messages or/and delivery of values will be updated. This method is called backward learning update.

The entries of the routing table, $\langle R^t \rangle$, and the active node table,$\langle R^a \rangle$, are very much dependent on router connectivity and node failure. Therefore we define a configuration, consisting of router network where routers are capable to handle the node failure and a system, over which the reduction semantics is defined.

Therefore the reduction semantics of the systems is described with respect to the router connectivity and node status. It is defined as a binary relation between constructs called configurations. A configuration is defined in terms of network of routers where routers are capable to deal with node failure and a system. The network of routers Γ_c is basically a binary relation between the router names. The configuration $\Gamma_c \triangleright$ S is taken from [6] and restated here as:

Configuration: 2.1 *A configuration consists of a pair (Γ_c , S) where $\Gamma_c \subseteq RN \times RN$, the network of routers connectivity, and RN is the set of router names in any system S. The undirected graph (RN, Γ_c) is connected.*

Notation: 2.1 *In a router connectivity Γ_c, for routers R_i,R_{i+1}, . . . , R_{k-1},R_k,*

$$R_i \rightsquigarrow R_{i+1} \rightsquigarrow \ldots \rightsquigarrow R_{k-1} \rightsquigarrow R_k$$

means {(R_i,R_{i+1}), . . . , (R_{k-1},R_k)} $\in \Gamma_c$. We shall use the notation p(R_i,R_k) to represent a path of routers, $R_i \rightsquigarrow \ldots \rightsquigarrow R_k$ between pair of routers R_i and R_k.

Further in DR$^{\omega}_{\pi}$F, for the purpose of determining the path between the communicating processes on node failure it is required that each router maintains two tables: a routing table which is represented by $\langle R^t \rangle$ and a active node table which is represented by $\langle R^a \rangle$. We can represent router table , $\langle$ R$\rangle$, as the combination of routing table, $\langle R^t \rangle$ and a active node table, $\langle R^a \rangle$. Before we formally define the routing table we shall describe the notation used for adjacent routers. We use the notation Adj(R) to represent the set of adjacent routers of R in Γ_c. It is defined as:

Adj(R) = $\{R' \mid (R,R') \in \Gamma_c\}$

Now formally, the router table, $\langle$ R$\rangle$, is defined as follows:

Definition: 2.1 *In a configuration $\Gamma_c \triangleright \langle R \rangle [\![N]\!] \mid S$, the named router table $\langle R \rangle$ is the combination of two tables: one is routing table, $\langle R^t \rangle$ and second is active node table, $\langle R^a \rangle$. At some router R, following two functions have been performed :*

1. *routing table, $\langle R^t \rangle$ performs the function, f: NN $\rightarrow$ RN , from the set of node names, NN, to the set of adjacent router names, RN. Formally expressed as $\langle R^t_i \rangle$ (r) = R_j, (R_i ,R_j) $\in \Gamma_c$.*
2. *active node table, $\langle R^a \rangle$ performs the boolean function, f_b: M $\rightarrow$ {T,F}, where M is the set of node names such that m $\in$ NN and present at R ; which returns T if node is active at router R(i.e. node is present in active node table $\langle R^a \rangle$); this is formally represented as $\langle R^a \rangle$ [m]$\downarrow$ or returns F if node is not active at router R (i.e. node is absent in active node table $\langle R^a \rangle$) ; this is formally represented as $\langle R^a \rangle$ [m]$\uparrow$.*

The function f can be many-to-one as many node names may have the same adjacent router on the path

towards the destination. But each destination node will have only one choice of adjacent router.
If a node name m belongs to the domain of the routing table $\langle R^t \rangle$ then we use the notation $\langle R^t \rangle$ (m)$\downarrow$. Similarly if the node name m does not belong to the domain of the routing table $\langle R^t \rangle$ then we use the notation $\langle R^t \rangle$ (m) $\uparrow$.

3 EXAMPLE

Now we will take an example to demonstrate the new node creation and message forwarding in DR^{ω}_{π}F. We will also demonstrate that how this new node name is propagated across the network and various routing table and active node tables entries are updated by this new node name using backward learning updates. This example also considers the cases of node failures and its semantic handling in the calculi.

Example: 1 *Let us take a configuration $\Gamma_c \ \triangleright S_1 \mid S_2 \mid S_3 \mid S_4$ where Γ_c = (R_1,R_2), (R_2,R_3), (R_3,R_1) as shown in the Figure 7. S_1, S_2 and S_3 are defined as follows:*
$S_1 \equiv \langle R_1 \rangle$ ⟦ *n[new node r with P in Q]* ⟧
$S_2 \equiv \langle R_2 \rangle$ ⟦ N_2 ⟧
$S_3 \equiv \langle R_3 \rangle$ ⟦ *m[M]* ⟧
$S_4 \equiv \langle R_4 \rangle$ ⟦ ⟦ *o[c?(x)R]* ⟧

where P≡ $o!\langle r@c \rangle \mid Q'$.
The routers connectivity Γ_c is defined as
{(R_1,R_2),(R_1,R_3),(R_1,R_4),(R_2,R_5),(R_3,R_5),(R_4,R_5)}
In this example a process at node n at router R_1 creates a new node r and this new node is exported by another process running at same node to a process at distant node o at router R_4. We shall see how various routing tables are updated about this new node using backward learning approach. With an application of the rule (R-NEWNODE -CREATION-ACTIVATION) in Figure 6 the configuration $\Gamma_c \ \triangleright S_1 \mid S_2 \mid S_3 \mid S_4$ can do a reduction to

$$\Gamma_c \ \triangleright (new\ r)(\langle R^t_1\ \{r \to R_1\}^+, R^a_1\{r\}^+ \rangle [\![r[P] \mid n[Q]]\!]\ ) \mid S_2 \mid S_3 \mid S_4$$

where a new node r has been created at router R_1 with process P in it. The routing table R^t_1 and active node table R^a_1 are also updated with the entries $\{r \to R_1\}^+$ and $\{r\}^+$ respectively because R_1 is the home for node r. Also note that no other router in Γ_c know about new node r.

Now the process $P \equiv o!\langle r@c \rangle \mid Q'$ at node r exports a value r which is a node name to another process at node o at router R_4. For convenience of writing terms we replace definitions of S_1, S_2 . . . where ever necessary because we know by the rule (R-STRUCT) that reductions are defined upto structural equivalence. Also we will use various axioms of structural equivalence in Figure 2, Figure 3, Figure 4 and Figure 5 to express the terms upto structural equivalence but don't explicitly mention them. An application of rule (R-OUT-NF) and (R-CONTX) will reduce the configuration

$$\Gamma_c \ \triangleright (new\ r)(\langle R^t_1\ \{r \to R_1\}^+, R^a_1\{r\}^+ \rangle [\![r[o!\langle r@c \rangle \mid Q'] \mid n[Q]]\!]\ ) \mid S_2 \mid S_3 \mid S_4$$

to

$$\Gamma_c \ \triangleright (new\ r)(\langle R^t_1\ \{r \to R_1\}^+, R^a_1\{r\}^+ \rangle [\![r[Q'] \mid n[Q]]\!] \mid [R_1]M^0_{sg}(r,\ o,\ r@c)) \mid S_2 \mid S_3 \mid S_4$$

Now the message $[R_1]M^0_{sg}(r,\ o,\ r@c)$ hops towards destination o for delivering the value. Suppose $\langle R^t_1 \rangle$ (o) = R_2. We know that (R_1,R_2) $\in \Gamma_c$. Since r is a new node therefore $\langle R^t_2 \rangle$ (r)$\uparrow$ is true, therefore with an application of rule (R-MSG - FWD-NF - I) and (R-CONTX) the message $[R_1]M^0_{sg}(r,\ o,\ r@c)$ hops at R_2. So the configuration

$$\Gamma_c \ \triangleright (new\ r)(\langle R^t_1\ \{r \to R_1\}^+, R^a_1\{r\}^+ \rangle [\![r[Q'] \mid n[Q]]\!] \mid [R_1]M^0_{sg}(r,\ o,\ r@c)) \mid S_2 \mid S_3 \mid S_4$$

reduces to

$$\Gamma_c \ \triangleright (new\ r)(\langle R^t_1\ \{r \to R_1\}^+, R^a_1\{r\}^+ \rangle [\![r[Q'] \mid n[Q]]\!] \mid [R_2]M^{0+1}_{sg}(r,\ o,\ r@c) \mid \langle R^t_2\ \{r \to R_1\}^+ \rangle [\![N_2]\!]\ ) \mid S_3 \mid S_4$$

Note that the routing table $\langle R^t_2 \rangle$ is updated with a new entry $\{r \to R_1\}^+$ because the message carrying r has hoped from R_1 and $\langle R_1 \rangle$ (r)$\downarrow$. The cost of this reduction is 0. Further suppose $\langle R^t_2 \rangle$ (o) = R_3 and message $[R_2]M^1_{sg}(r,\ o,\ r@c)$ is propagated to R_3. Since $\langle R^t_3 \rangle$ (r)$\uparrow$ and (R_2,R_3) $\in \Gamma_c$ therefore again using the rule (R-MSG-FWD- NF-I) and (R-CONTX) the configuration

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_2]M_{sg}^{0+1}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket\ ) \mid S_3 \mid S_4$$

is reduced to

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_3]M_{sg}^{2}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_2\}^+ \rangle \llbracket m[M] \rrbracket\ ) \mid S_4$$

Note that the routing table $\langle R_3^t \rangle$ *is updated with* $\{r \rightarrow R_2\}^+$ *because the message* $[R_3]M_{sg}^2(r,\ o,\ r@c)$ *has hoped from* R_2 *and* $\langle R_2^t \rangle (r)\downarrow$ *. Further suppose* $\langle R_3^t \rangle (o) = R_4$ *and message* $[R_3]M_{sg}^2(r,\ o,\ r@c)$ *is propagated to* R_4*. Since* $\langle R_4^t \rangle (r)\uparrow$ *and* $(R_3,R_4) \in \Gamma_c$ *therefore again using the rule (R-MSG-FWD- NF-I) and (R-CONTX) the configuration*

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_3]M_{sg}^{2}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_2\}^+ \rangle \llbracket m[M] \rrbracket\ ) \mid S_4$$

is reduced to

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_4]M_{sg}^{3}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_1\}^+ \rangle \llbracket m[M] \rrbracket \mid \langle R_4^t \{r \rightarrow R_3\}^+ \rangle \llbracket o[c?(x)R] \rrbracket\ )$$

Note that the routing table $\langle R_4^t \rangle$ *is updated with* $\{r \rightarrow R_3\}^+$ *because the message* $[R_4]M_{sg}^3(r,\ o,\ r@c)$ *has hoped from* R_3 *and* $\langle R_3^t \rangle (r)\downarrow$ *. The cost of this reduction is 3. Before proceeding further it is worth noting that routers* R_2 *,*R_3 *and* R_4 *are updated with relevant entries about the new node r in the process of delivery of the value r from a process at source node r at* R_1 *to a waiting process at node o at* R_4*. All the routers in path of communication between* R_1 *and* R_4 *are updated. But if there are many other routers then they don't know about the new node name. They will not know about the new node name until it is propagated along any path. This method of routing table update is known as backward learning.*

Because $\langle R_4^t \rangle (o) = R_4$*, now there are two cases:-*

Case 1: *if active node table,* $\langle R_4^a \rangle$ *, function* f_b *returns T i.e. node o is active at router* R_4 *i.e* $\langle R_4^a \rangle [o]\downarrow$ *.*

Case 2: *if if active node table,* $\langle R_4^a \rangle$ *, function* f_b *returns F i.e. node o is not active at router* R_4 *i.e* $\langle R_4^a \rangle [o]\uparrow$ *.*

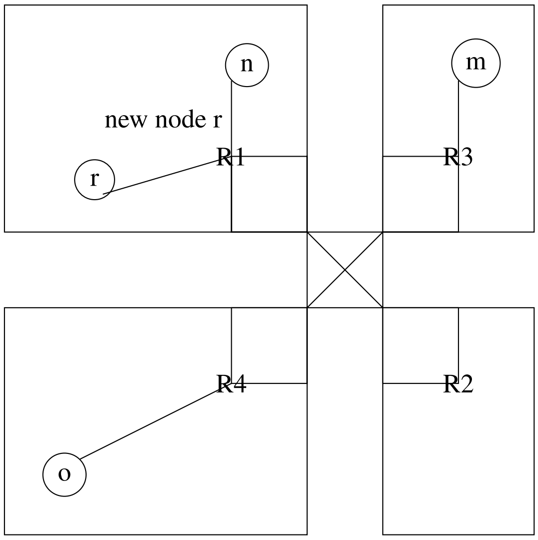

Figure 7: A Distributed Network of Routers

In case 1 the node o is active at router R_4*, so the value r is delivered to the waiting process at o using the rule (R-IN-NF-I). Therefore the configuration*

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_4]M_{sg}^{3}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_1\}^+ \rangle \llbracket m[M] \rrbracket \mid \langle R_4^t \{r \rightarrow R_3\}^+ \rangle \llbracket o[c?(x)R] \rrbracket\ )$$

reduces to

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_4]M_{sg}^{3}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_1\}^+ \rangle \llbracket m[M] \rrbracket \mid \langle R_4^t \{r \rightarrow R_3\}^+ \rangle \llbracket o[R\{^r/x\}] \rrbracket\ )$$

In case 2 the node o is not active at router R_4*, so using (R-IN-NF-II) the value r in the message is not delivered at channel c. Now with an application of the rule (R-IN-NF-II) the configuration*

$$\Gamma_c \triangleright (new\ r)(\langle R_1^t \{r \rightarrow R_1\}^+, R_1^a\{r\}^+ \rangle \llbracket r[Q'] \mid n[Q] \rrbracket \mid [R_4]M_{sg}^{3}(r,\ o,\ r@c) \mid \langle R_2^t \{r \rightarrow R_1\}^+ \rangle \llbracket N_2 \rrbracket \mid \langle R_3^t \{r \rightarrow R_1\}^+ \rangle \llbracket m[M] \rrbracket \mid \langle R_4^t \{r \rightarrow R_3\}^+ \rangle \llbracket o[c?(x)R] \rrbracket\ )$$

reduces to

$$\Gamma_c \triangleright (new\ r)(\langle R^t_1 \{r \rightarrow R_1\}^+, R^a_1\{r\}^+\rangle [\![r[Q'] \mid n[Q]]\!] \mid \langle R^t_2 \{r \rightarrow R_1\}^+ \rangle [\![N_2]\!] \mid \langle R^t_3 \{r \rightarrow R_1\}^+ \rangle [\![m[M]]\!] \mid \langle R^t_4\{o \rightarrow R_4\}^-, R^a_4\{o\}^-\rangle [\![ ]\!])$$

Note the updated entries at routers R_1,R_2,R_3 and R_4 as a result of these reductions. With this example we complete the discussion of reduction semantics.

4 CONCLUSION

In this paper we have discussed a new language $\mathrm{DR}^{\omega}_{\pi}$F. In $\mathrm{DR}^{\omega}_{\pi}$F we have extended the language $\mathrm{DR}^{\omega}_{\pi}$[5] to handle node failure during communication. At the primitive level we have discussed only syntax and semantics of the calculi. $\mathrm{DR}^{\omega}_{\pi}$F contains only semantic extension of $\mathrm{DR}^{\omega}_{\pi}$[5], therefore the proof systems of equivalence between specification and its implementation should directly apply to $\mathrm{DR}^{\omega}_{\pi}$F as well. We further propose to work towards wellformedness and name forms full abstraction proof the calculi.

REFERENCES

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

[2] editor Douglas Comer. *Internetworking with TCP/IP - Principles, Protocols, and Architectures,*. Prentice-Hall, fourth edition, 2000.

[3] Adrian Francalanza and Matthew Hennessy. A theory of system behaviour in the presence of node and link failures. *In CONCUR*, pages 368–382, 2005.

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

[5] Manish Gaur. A routing calculus for distributed computing. *In Elena Troubitsyna, editor, Proceedings of Doctoral Symposium held in conjunction with Formal Methods 2008,Turku Centre for Computer Science General Publication*, 48:23–32., May 2008.

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

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

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

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

[10] James F. Kurose and Keith W. Ross. *Computer Networking: A top down approach featuring the Internet.* Addison Wesley, 2001.

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

[12] Robin Milner. A calculus of communicating systems. *volume 92 of Lecture Notes in Computer Science. Springer*, 1980.

[13] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

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

[15] Benjamin C. Pierce and Davide Sangiorgi. Typing and subtyping for mobile processes. *Mathematical Structures in Computer Science*, pages 6(5):409–453, 1996.

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

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

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

[19] Andrew S. Tanenbaum. *Computer Networks.* Pearson Education, Inc.,, Upper Saddle River,New Jersey,, fourth edition, 2003.