EPCglobal Network Distributed Discovery |Services using extended version of BGP

Mazen G. Khair, Hussein T. Mouftah, *Fellow IEEE*

Abstract— **In this paper, we propose a distributed Discovery Services approach for the EPCglobal networks using Electronic Product Code-Border Gateway Protocol (EPC-BGP). Each EPCglobal network will be equipped with a gateway that connects this network to the internet in which all EPCglobal networks will be connected to each other. Each gateway will make use of the EPC-BGP to exchange routing information about the shipped objects. Each gateway keeps a list of EPCs of all shipped objects to other EPCglobal networks. This information can be used to exchange any new information available about any object for example product recall.**

EPC-BGP exploits most functionalities of the BGP routing protocol with some modifications to support the exchange of routing information about EPCs. Furthermore, EPC-BGP introduces an advertising policy to allow the gateway to advertise routing changes based on the number of changes which took place in the EPCIS table. We have defined three types of blocking: Justified Update Refusal, Unjustified Update Acceptance and Unjustified Update Refusal. The performance of this protocol for different network architectures has been investigated through simulation studies, and the results show promise.

***Index Terms*— BGP, Discovery Services, EPCglobal, RFID**

I. Introduction

The term Internet of Things (IoT) was first introduced in the EPC Global standards in 1999 [1]. It is intended to extend the Internet from a network of computers to a network of things (or objects). To make this possible the Electronic Product Code (EPC) has been introduced, the EPC is stored in a Radio Frequency Identification (RFID) in which it can be read through RFID reader. The RFID is attached to an object to allow tracking this object across the supply chain.

The EPCGlobal network is a consortium of companies and organizations set up to achieve worldwide standardization and adoption of RFID. The EPCglobal network consists of three major components: The client application, the Object Naming System (ONS) root and the EPC Information Services (EPCIS). The way it works is as follow: The client application that acts as middleware receives the Electronic Product Code (EPC) from the RFID reader and pass the EPC to the Object Naming System (ONS) after converting the EPC to a Uniform Resource Identifier (URI) that can be realized by ONS. The ONS functions like a "reverse phone directory" since the ONS uses a number (EPC) to retrieve the location of EPC data from its databases. The ONS return the Uniform Resource Locator (URL) that can be used by the client application to retrieve the information about the corresponding EPC from the EPC IS data server. To extend the internet of things definition, the EPCglobal network should be connected to the internet through a gateway in which all EPCglobal networks can be connected to the internet. This means that each physical object can be reached through the internet. In this paper , we propose a distributed architecture of a discovery services in which connects different EPCglobal networks together through the internet to allow product recall, each EPCglobal network will have a gateway that connects this EPCglobal network to the internet. This architecture will allow different EPCglobal networks to share information about any physical object that has an RFID tag. If new information about a physical object becomes available this information will be propagated to other EPCglobal networks through the gateway. We have defined a new protocol known as EPC-BGP protocol that exchange information about physical objects among different EPCglobal networks. This new architecture will allow each EPCglobal network to share information about its objects if new information is available such as the need for product recall.

M. Khair, and H. T. Mouftah are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario K1N 6N5, Canada (e-mail: mkhair, mouftah@site.uottawa.ca).

II. Related Work

In [2], Beier et al have been the first to present a Discovery Services summarized as Directory Look-up approach. In their work, they come to a conclusion that ONS is not suitable for developing a discovery services. Their approach works as follows: objects that holds RFIDs that holds EPCs travel through the supply chain. At each step through the supply chain, the RFID tag will be read and the information will be stored in the EPC-IS of that step. If this EPC has been stored in this location for the first time, the Discovery Service is notified by sending the EPC, the URL of that EPC in the corresponding EPC-IS, the timestamp and for security the certificate of the submitter, and finally, a visibility flag in its repository. In their implementation, they assume that all participants of all EPC networks are equipped with a certificate by a trusted third party. Hence, the Discovery Service can be queried with the EPC of interest. The Discovery Service replies back with the relevant EPC-IS URL

in which allows the requester to contact all relevant EPC-IS to get the desired information about the EPC of interest.

In [3], the authors collect information about an item following the 'E-Pedigree' model. All the EPCs of the objects being shipped are listed in the EPedigree document. The way it works is as follow, EPedigree document is initially created and encrypts by the manufacturer and pass it on the next entity in the supply chain, the next entity perform the same procedure by adding his data and electronically signs the document and then pass it to the next entity in the supply chain. The same process keeps going like this across the entire supply chain. Depending on retrieving which end of the supply chain, the authors propose a distributed architecture for forward tracing or for reverse tracing through the supply chain. Tracing up the supply chain can be done by using the General Manager Number stored in an EPC [4] in along with the Object Naming Service (ONS) [5] which is able to resolve the General Manager Number to the respective EPCIS. This approach uses distributed ONS architecture based on distributed hash tables. The retrieved EPC-IS address using the distributed ONS architecture is used to traverse the entire supply chain to locate information about specific EPC. They define this approach as 'Daisy Chain'. Since their approach is distributed, they claim that it is scalable for item-level product tagging. However, there is a disadvantage that they rely on re-writable tags.

BRIDGE stands for Building Radio frequency IDentification for the Global Environment [6]. This is an EU-funded project aims to deploy the EPCglobal network in Europe through researching, developing and implementing tools to achieve this goal. In their work, they select four out of eight proposes of Discovery Services approaches after careful evaluation. Before we continue describing the four favorite approaches, it is worth to mention that EPC-ISs can serve two different types of queries. The first one is One-off queries, this one is performed by a client once and no further communication between client and EPCIS is planned, Whereas, the second one is Standing queries in which two mode of operation can be done here. First one can be time controlled using a query scheduler, for example, a requester prefer to be informed every hour. The second one is trigger-controlled, for example, the client will be informed based on the availability for new information about the EPC of interest [7]. What follows, we will describe the four preferred Discover Services approaches: The first approach is identical to the Directory Lookup approach [2] and is known as Directory-of-Resources. The second approach works exactly like the first approach except that a client shows interest about certain information by creating a subscription at the Discovery Service, this approach known as Notification-of-Resources. The third preferred approach is known as Notification of Clients, in this approach the EPC-ISs provide any new information about a certain EPC to the discovery services. On the other hand, if a client shows interest in a specific EPC, the Discovery Server informs all relevant EPCISs. The way it works is that the EPC-IS sends a notification to the clients informing that there is new information about the EPC of interest, on return the client queries the respective EPCISs. The last one is known as Query Propagation and acts like third approach except that the information is sent to the client by the EPCISs immediately without the availability notification.

In the related work section of [8], the authors criticize the approach of solving the Discovery service problem using the Domain Name Service and the Service Location Protocol as not appropriate. The authors in [8] criticize the approach in [2] from a security point of view, The authors in [8] believe that due to the fact that there are no access control policies in place, the address of EPCIS of that company that holds information about the EPC of interest will be reviled. They also explained that if these policies were implemented, this will lead to more complexity and requires more maintenance effort. This happen due to the fact that fine-grained access rights would have to be defined at Discovery Service level and policies would have to be synchronized between companies' EPCISs and the Discovery Service. To achieve the desired requirement which is low complexity and low access right maintenance effort, the authors in [8] present Query Relay approach presented in details in [6]. The advantage they present in their approach is that the EPCISs reply directly to the requester and the Discovery Service is the responsible of relaying the requests by forwarding the respective client queries to all relevant EPCIS. This means that EPCIS address is not revealed to the requester if the respective company decides not to reply to the query at all.

In [9], the authors present a follow up for the previous BRIDGE project [6]. Besides describing the implementation of the Directory Service approach, The authors discuss several design aspects such as using Light weight Directory Access Protocol (LDAP) that allow efficient data storage. It also describe interfaces of the Discovery Service adhere to the EPCIS standard[7]. Moreover, it describe an integration module within EPCISs that works as a sniffer for read events that are relevant for being published to the Discovery Service and finally, it describes an implementation for security based on using X.509 certificates to build a public key infrastructure between EPCISs and Discovery Service.

In [10], the authors propose an Aggregating Discovery Service (ADS). ADS is responsible for both forwarding the client queries to relevant EPCISs and aggregating the EPCIS responses to the client request after synchronizing the responses. By doing so, client complexity will be reduced and achieve low response latency. Furthermore, it delivers full and correct information for the requester and avoids the need for fine-grained access control replicated at Discovery Service level. And finally it guarantees confidentiality of clients. ADS is a centralized services that provide two interfaces. The first one is Notify interface whereas the second one is Query interface. Regarding the first one, it is used to inform the ADS about read events to be shared within the EPCglobal network. The ADS receives the EPCIS URL of the submitting partner and one or more EPCs that have been handled by this entity. The ADS maintains an association between submitting EPCISs and submitted EPCs. This allows the ADS to determine all EPCISs that hold more information about an EPC. Regarding the second interface, it provides similar approach detailed in [9] for querying the EPCIS [7]. The way it works is as follow,

once the client query arrives at the ADS, it parses the query to extract relevant EPCs, the ADS then make use of the EPCs using its internal data base to look up the URL of EPCISs which are relevant for this query, once ADS looks up the URL, it forwards the original query to those EPCISs. Each EPCIS replies back with the information about the EPCs of interest, the ADE aggregate the results and return the results back to the client. Clearly, the ADS acts as a proxy that is transparent to the client

III. Inter-domain Routing

A. Border Gateway Protocol

The *Border Gateway Protocol (BGP)* is an external routing protocol. BGP is a Path Vector (PV) type protocol. In PV, each border router advertises the destinations it can reach to its neighboring Edge Routers along with the information that describes various properties of the paths to these destinations. In other words, PV defines the route as a pairing between the destination and the attributes of the path to reach that destination. Thus the name path-vector comes from the fact that each edge router receives from its neighboring edge router a vector that contains a set of routes [11].

BGP is used to exchange routing information among different Autonomous Systems (ASs) located in different geographical regions [12]. The main task of BGP is to allow edge routers that belong to a certain AS to exchange network reachability information with other BGP systems located in different ASs [13]. This network reachability information includes information on the sequence of ASs that the reachability information passes through.

This information is used by edge routers to build a map of how ASs are connected to each other. Besides, each AS can make use of this routing information to detect and prevent routing loops. Loops can be detected because each AS can reject any routing information that has its AS name in it. Furthermore, BGP can give each AS the ability to apply a certain routing policy by allowing each AS to set the cost of the advertised link or prevent certain routing information to reach certain costumers [14].

Since BGP is a well-known protocol for inter-domain routing, we decided to adopt this protocol to achieve inter-domain routing among EPCglobal ASs and to make use of its routing features rather than to implement something from scratch.

BGP has been deployed in the Internet for a long time and it has the following features [14], [12], [13]:

- BGP functionality based on DV algorithms that tries to find the minimum number of hops to the desired destination [14].
- BGP relies on TCP as its carrier protocol with few modifications to achieve security [15]. The idea is that to every packet in a TCP session a field is added with the MD5 checksum of the packet contents and a secret key. This establishes a cryptographically secure signature of the packet. Without knowing the key, it is near impossible to construct a packet with a valid signature. Since BGP speakers will immediately discard packets without a signature or with an invalid signature, all types of attacks cannot be executed without knowing the key
- BGP has the ability to apply policy-based control; because BGP is manually configured, therefore each BGP router can decide which received addresses can be accepted. Besides, it can prevent some addresses propagateing to certain customers [12].
- Usually, edge routers that speak BGP advertise only one prefix to one of its BGP neighbours within an update message. Normally, the advertised address/prefix is associated with a number of attributes such as Cost, Next_hop, As_path –etc [13]. We will discuss these attributes later in this section.

Figure 1 shows a simple scenario for a network that has multiple ASs connected together. Clearly, there are five ASs, each has a number of edge routers. This scenario also shows the geographical working area for each routing protocol; intra-domain routing protocol like Open Shortest Path First (OSPF) and inter-domain routing protocol like BGP.

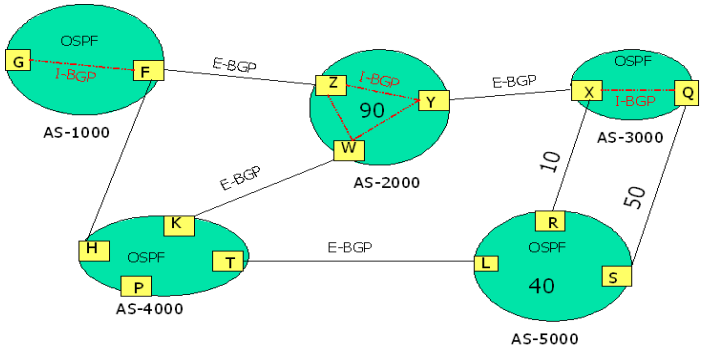

Figure 1: A network scenario consisting of five ASs

During a BGP session between any two edge routers, the two participating peers exchange update messages. Each update message includes one prefix associated with a certain number of attributes such as AS_Path, Next_hop, cost --etc [14] [13].

BGP Message Types

BGP has four types of messages that share the same header structure. The four messages are: Open message, Notification message, Update message and finally Keep alive message [15].
Open message: is the first message that is used by any edge router after the TCP connection has been established. Usually, both participants send each other this message to identify each other [14].
Update message: this message is used to add or remove a prefix. For example, if a certain prefix no longer exists then the BGP update message will include this prefix in the withdraw section whereas if a new route "prefix" has been discovered then this prefix will be advertised as a new route [14].
Notification message: this message is used to report a fault during a BGP session [14].
Keep alive message: this message is sent during a BGP session between two BGP speakers to confirm that the BGP session is

still alive [14]. The BGP session is kept alive until the two participants exchange their new routes.

BGP Path Attributes

BGP attributes are very important for any advertised route because these attributes allow the receiving BGP system to understand the reachability information for the received path [15]. Although, there are only seven path attributes defined by IETF in RFC1771, many new attributes are defined by other RFC documents to enhance BGP to support other functionalities.

In this section, we shall only discuss the seven attributes that are documented by the RFC1771. Usually, a certain number of these attributes are exchanged by a BGP update message to describe a route. The seven attributes are:

1- Origin Attribute: The Origin attribute can have three values: 0, 1 or 2. If the value of the Origin attribute is 0, then the Network Layer Reachability Information (NLRI) is received via an Interior routing protocol such as OSPF. If the Origin attribute has a value of 1, then the NLRI learned via an exterior routing protocol such as BGP [14]. Finally, if the Origin attribute has a value of 2, then the NLRI learned via a different means. It could be set for example by the network administrator [15].

2- AS_Path Attribute: The AS_Path attribute is essential to prevent loops that may occur among ASs [14]. To understand the benefit of this attribute, we will consider the example in figure 1. Assume that AS_3000 in Figure 1 has a BGP session with AS_2000. This session will allow AS_2000 to construct a route to AS_3000. In turn, AS_2000 can have another BGP session with AS_4000 to pass on the route that was learnt from AS_3000. As a result, AS_4000 will get enough information to enable it to reach AS_3000 through AS_2000. Now AS_4000 performs the same task to pass the AS_Path information to AS_5000. If AS_5000 decided to advertise the route obtained by AS_4000 to AS_3000, AS_3000 would not accept the route because AS_3000 could easily detect that this route was originated by itself and hence, a routing loop is prevented.

3- NEXT_HOP Attribute: Usually, the value of this attribute is the virtual interface1 or loopback interface if there is an internal session to exchange routing information among edge routers that belong to the same AS, whereas this attribute takes the IP address of the physical link of the source edge router within an External BGP session with the neighbor AS.

4- MULTI_ EXIT_DISC Attribute: This attribute is used when there is more than one connection between two adjacent ASs. This value can assist with choosing the optimal path across this AS [13]. For example, Figure 1 shows that AS_5000 is connected with AS_3000 via two connections. AS_5000 can give different values for each connection when it advertises its routing information to AS_3000. Assume that AS_5000 will give a value equal to 10 for the connection between router S & Q, while it gives a value equal to 50 to the connection between router R & X. These values will guide AS_3000 to select between the two links. AS_3000 will choose the connection that has lower MULTI_ EXIT_DISC value. In this example, AS_3000 would choose the X-R connection because it has a value equal to 10, which is lower than the S-Q connection, which has a value of 50.

5- Local_Pref Attribute: there could be many possible ways to cross a given AS. This attribute can be used locally by each AS to control the path for crossing that AS. In other words, which edge routers that belong to the same AS will be used to transport the traffic [13]. For example, Figure 1 shows that Router Y in AS_2000 can allow Router X located in AS_3000 to reach Router F located in AS_1000 through two different paths. The first path could be through Y, Z whereas the other could be through Y, W and Z. In fact, the Local_Pref attribute allows edge routers Y, Z, and W that belong to AS_2000 to agree among themselves on the path that should be used to handle a certain path request.

6- Atomic_Aggregator Attribute: When two routes have overlapped at a certain edge router, this router can use this attribute to tell neighbor edge routers about this route overlapping [15].

7- Aggregator Attribute: It is necessary to aggregate more than one prefix into a single prefix for reason of scalability. This attribute indicates the AS and the router that perform the aggregation. For example, AS_2000 can aggregate the addresses of three ASs,(AS_3000, AS_4000, and AS_5000), into one address/prefix. This new aggregated address can be advertised to AS_1000.

B. BGP Operation "E-BGP and I-BGP"

When BGP is used between two different ASs, this mode of operation is referred to as External BGP (E-BGP) [15]. If an Internet Service Provider is using BGP to exchange routes that have been learned by an external BGP session or by any other means within its AS, then this mode of operation is referred to as Internal BGP (I-BGP) [15]. The most important fact about the operation of BGP as I-BGP is that each node has to peer with all other edge nodes located in the same AS through a logical connection [13]. The reason for these logical connections is to allow edge routers that belong to the same AS to exchange the routing information learned via different external sources. This mode of operation is known as "full-mesh I-BGP" [12]. In fact, since I-BGP sessions are logical sessions, there are no direct physical connections among the participants. Each AS configures these logical connections among its edge routers. For example in Figure 1, node Y in AS

[1] Virtual interface is an address used for identifying a router, it has no relation with any physical or hardware interface. This address is very useful for performing Intra-Domain routing when there is no direct physical connection between edge routers that belong to the same AS.

2000 will have a logical peer with both edge routers Z and W. These logical paths are configured by the network administrator of AS_2000, the network administrator will specify the intermediate nodes that will be used to establish the I-BGP session between router Y and Z and the intermediate nodes that will be used to establish the I-BGP session between router Y and W.

IV. EPCGLOBAL NETWORK ARCHITECTURE

A. *EPCglobal Network*

Figure 2 shows the EPCglobal network, it shows all the steps from reading the tag up to viewing the information about the object the tag is attached to. The process starts with reading the tag using RFID reader.

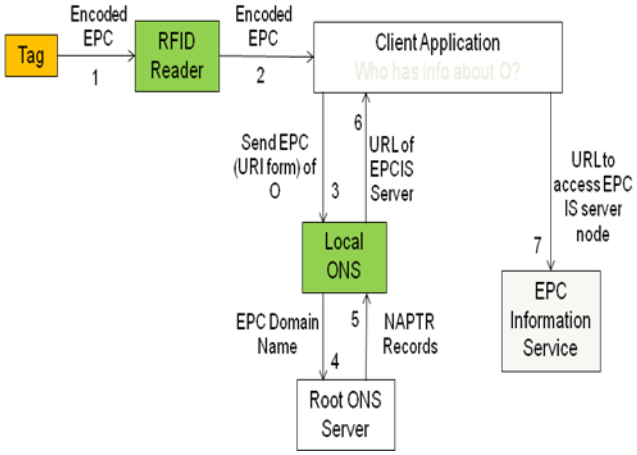

Figure 2: EPCglobal network

The RFID reader reads the Tag and gets the EPC and presented it to the Client application, figure 3 shows an EPC example, it shows the EPC for SGTIN-96 example.

Header 8 Bits	Filter 3 Bits	Partition 3 Bits	Company Prefix 20-40 Bits	Indicator/Item ref 20-40 Bits	Serial Number 38 Bits
00110000 "SGTIN-96"	001 "Retail"	101 "24:20 Bits"	200452	5742	5508265

Figure 3: Electronic Product Code (SGTIN-96 Example)

There are seven GS1 keys in which each of them is corresponding to an EPC, STGIN is one of them. The rest are: SSCC, GLN, GIAI, GSRN used to identify unique objects. GRAI, GDTI: a hybrid that may identify either categories or unique objects depending on the absence or presence of a serial number. And Finally, there are two other keys, GINC and GSIN: identify logical groupings, not physical objects. Regarding the SGTIN-96 example, each field represent the following: Header: followed by a series of fields whose overall length, structure, and function are determined by the header value. Filter: allow an RFID reader to select or deselect the tags corresponding to certain physical objects, for example in a pallet or container. Partition: contains a code that indicates the number of bits in the GS1 Company Prefix field and the Indicator/Item Reference field. Company Prefix: is assigned by GS1 to a managing entity. Item Reference: is assigned by the managing entity to a particular object class. Serial Number: assigned by the managing entity to an individual object.
This SGTIN-96 will be presented to the client application as bit sequence and the client application convert the sequence of bits as shown in this example : urn:epc:id:sgtin:200452.5742.5508265. The client application server present this to the local Object Naming Service (ONS) [16], [17] as shown in figure 2, step 3. The ONS convert the EPC to the following format "5742.200452.sgtin.id.onsepc.com" before sending it to the ONS root as step 4. The ONS root returns a series of answers that contain URLs that point to one or more services (for example, an EPCIS Server), step 5. Finally, depending on the service that the ONS Client desires, it uses one or more of the NAPTR records returned to locate an appropriate service

Order	Pref	Flags	Service	Regexp	Replacement
0	0	U	EPC+epcis	!^.*$!http://example.com/cgi-bin/epcis!	.(a period)

Figure 4: NAPTAR Record

Figure 4 shows a NAPTR record, A Name Authority Pointer (NAPTR): a type of resource record used in the Domain Name System (DNS). The NAPTR record contains the following: The **Order** field is set to zero. The **Pref** field is set to a non-negative integer. The value of the Pref field is an ordinal that specifies that the service in one record is preferred to the service in another record having the same Service field. The **Flags** field is set 'u', indicating that the Regexp field contains a URI. The **Service** field contains an indicator of the type of service that can be found at the URI in question. The **Replacement** field is not used by the EPC Network but since it is a special DNS field its value is set to a single period ('.') instead of simply a blank. The **Regexp** field specifies a URL for the service being described.
Finally, the local ONS return the selected URL back to the client application in which in turns it will presented to the EPCIS data server to obtain information about object O.

B. *EPCglobal Discovery Services architecture*

Current EPCglobal network does not have the ability to track an object if it has left the boundary of its network. Normally, an object is manufactured by a certain company and transported and stored by another company and sold by another company. To trace all the information about this object, any EPCglobal network must be able to access the information about this object in another EPCglobal networks. For example, one should be able to get information about a drug in the pharmacy, who manufacture it, and where it has been stored and under which conditions. Therefore, if one wants to get all this information at the pharmacy, the EPCglobal network of the pharmacy must be connected some how to the EPCglobal network of the manufacturer and the warehouse where the drug has been stored. This means that the manufacture EPCglobal network and the warehouse EPCglobal network and the retailer or pharmacy EPCglobal

network must be connected. To achieve this, each EPCglobal network will have a gateway that connects that EPCglobal network to the internet. Figure 5 shows the proposed architecture of the EPCglobal network. The gateway becomes part of the discovery services in which it can allow to trace all the information about a given object as long as it has an RFID. If any object has new information, this information will be added in the EPC-IS data server. Depending on the number of changes in the EPC-IS data server, this information will be passed to the gateway which has routing information about the origin and the destination of this EPC. The gateway will look the EPC up and finds the path that identify the path of this EPC, the process will be explained later in the next section.

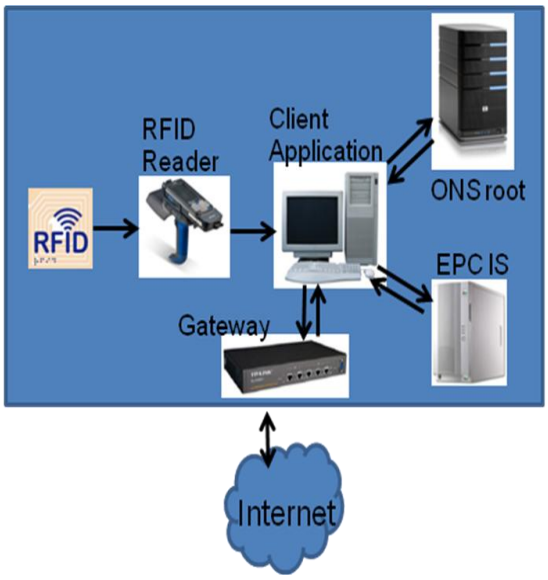

Figure 5: EPCglobal architecture with discovery services

V. EPC-BGP

As mentioned above, there are more than these seven attributes for BGP. [15] shows all the attributes that have been defined so far by IETF. In fact, some of these attributes could be used in routing among EPCglobals whereas others are redundant. Our work will focus on using some of these attributes to develop a routing protocol that will serve the needs of a EPCglobal network. We shall discuss the benefits of only some of these attributes in EPCglobal networks in this section. The purpose of this work is to define a routing protocol that can perform EPC exchange among EPCglobal networks.

A. EPC-BGP overview and goals

Electronic Product Code Border Gateway Protocol (EPC-BGP) is built on the experience gained from BGP. In fact, much research is going on to improve the operation and scalability of BGP which suggests a promising future for this routing protocol [18].

EPC-BGP adopts most of BGP's features with some differences that allows EPC information to be exchanged among edge routers to reflect the up-to-date status of the EPC. Furthermore, EPC-BGP introduces a new route advertising scheme which is triggered by the number of changes that took place in the EPCglobal network.

There are many reasons that led to the definition of a new routing protocol. One reason is to have a routing protocol that serves the EPCglobal's basic needs. Of course, their basic need is to have routing information at their edge routers that guarantees finding the most up-to-date information about any EPC. Therefore, the aim of EPC-BGP is to guarantee a reliable mechanism to exchange EPC among different EPCglobal networks to guarantee the most up-to-date information about each EPC.

B. EPC-BGP messages

As we explained earlier, EPC-BGP is built on the experience gained from BGP. In section 3, we explained BGP in detail. Besides, we explained the message types and the purpose of these messages that are being used in BGP [14]. In our approach, we are using most of the BGP features [12].

The EPC-BGP session starts when the advertising node sends an open message to identify itself. Once the BGP session has been established, the two edge nodes exchange the EPC information using the update message [14], [12], [13]. Both nodes keep the EPC-BGP session alive by sending the keep alive message. Finally, both nodes report errors by simply sending a notification message. In this section, we will only discuss the new information that has been added or replaced within the BGP update message to carry EPC information.

Figure 6 shows the structure of an EPC-BGP update message that is being exchanged. As explained earlier in section 3, each update message advertises a route using a certain number of attributes. These attributes are:

Figure 6: EPC-BGP Update Message structure

- Next_Hop attribute field: this field indicates the port number/address of the source edge router that leads to the next hop that will eventually lead to the desired destination. The port number/address represents a unique number of the physical link if there is an external EPC-BGP session with the neighbor EPCglobal network, whereas the next hop could be something like the *virtual interface* in BGP if there is an internal session to exchange routing information among edge routers that belong to the same EPCglobal network.
- As_path attribute field: this field includes the list of the EPCglobal network names that will lead to the desired destination.
- Cost attribute field: this is a new attribute that specifies the cost of the link being used.
- Originator attribute field: this field stores the name of the edge router originating this message. This value will be used by the receiving edge router to prevent

sending the same routing information back to the sender.

- Available EPCs: shows the available EPC that is being shipped.
- Destination field: stores the destination prefix of the destination EPCglobal network.

The EPC-BGP session will be kept alive until all the EPCs that is going to be shipped is being exchanged with the destination EPCglobal network. If necessary, more than one EPC-Update message will be exchanged until all the EPCs of the requested shipment has been exchanged.

C. Routing Table Structure

At each edge router of an EPCglobal network, there should be a routing table that shows the available EPC to all destinations. To simplify things, we considered the following network example shown in Figure 7. This small network has six EPCglobal networks that represents a mesh network with a diameter equal to 2. Each EPCglobal network has one edge router or gateway. We write $C_{i,j}$ for the cost of the link between the two edge routers of the EPCglobal networks i and j, as shown in the figure.

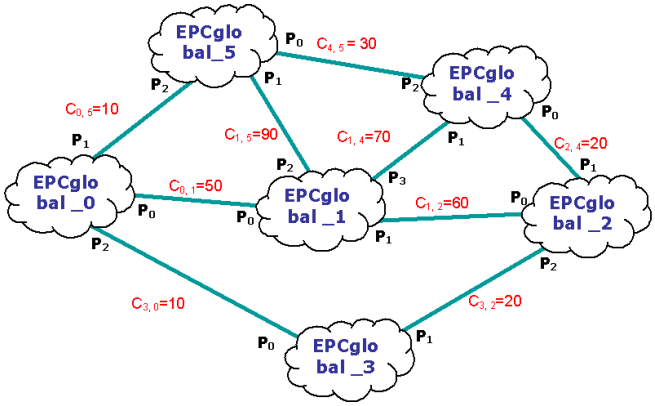

Figure 7: Simple network example

TABLE 1
ROUTING TABLE FOR NODE 0

Dest EPCglobal	NextHop	AS_Path	Cost	EPC status										Originator
0	*****	0	0	T	T	T	T	T	T	T	T	T	T	0
1	Port_0	1	50	T	T	T	T	F	T	T	T	T	T	1
2	Port_2	3 2	30	T	T	T	T	T	T	T	T	T	T	3
3	Port_2	3	10	T	T	T	T	T	T	T	T	T	T	3
4	Port_1	5 4	40	T	T	F	T	F	T	T	T	T	T	5
5	Port_1	5	10	T	T	T	T	F	T	T	T	T	T	5

Table 1 shows the routing table for EPCglobal_0. It includes all the necessary routing information that EPCglobal _0 should have to be able to exchange the status of each EPC with other EPCglobal network. This means when ever any more information about a certain EPC is available the status of that EPC will be changed to indicate that an exchange of this information with other EPCglobal networks is needed.

- The first column shows all the destinations that EPCglobal _0 can reach.
- The second column gives the port number/address of the link that should be used to reach the correspond destination in column number one.
- The third column shows the AS_Path attribute, the AS_path includes the list of the EPCglobal network names the routing information traversed.
- The fourth column is the cost of the path; the cost could be a function of the connection media between the two EPCglobal networks. Figure 7 shows that EPCglobal _0 can reach EPCglobal _4 through EPCglobal _5. Therefore, the cost between EPCglobal _0 and EPCglobal _4 is the summation of the cost along the path. In our example, the cost will be $C_{0,5} + C_{5,4} = 10 + 30 = 40$.
- The fifth column represents the status of the available EPCs that has been shipped to the corresponding destination in column 1.
- The sixth column represent a flag that indicates from were this route has been originated, for example EPCglobal _1 will set the originator attribute equal to 1 when EPCglobal _1 decides to advertise a route to EPCglobal _0. EPCglobal _0 will accept the advertised route if EPCglobal _0 finds that it is a useful route. Now, when EPCglobal _0 decide to advertise this route to other EPCglobal, the originator attribute will prevent EPCglobal _0 from sending this route back to its origin, which is EPCglobal _1.

D. EPC-IS Table

Each EPC-IS at each EPCglobal network has a table that keeps a record of the shipments to other EPCglobal networks, it has a list of all EPCs that has been shipped to other EPCglobal networks. Table 2 shows the table of the EPC-IS of EPCglobal_0 network.

TABLE 2
EPC-IS OF EPCGLOBAL_0 NETWORK

Dest AS	EPC Status									
1	T	T	T	T	F	T	T	T	T	T
2	T	T	T	T	F	T	T	T	T	T
3	T	T	T	T	F	T	T	T	T	T
4	T	T	T	T	F	T	T	T	T	T
5	T	T	T	T	F	T	T	T	T	T

This table will keep track of the EPCs status of the shipped objects to other EPCglobal networks, this table will be updated when a certain tag has been read. In other words, a new information about this object is available. So, the status of the updated EPC will be changed from True to False indicating that new information about the EPC is available. Whereas the status is changed from False to True if the table has been updated, meaning that this information has been shared with other EPCglobal networks.

E. Routing policies and Information processing among EPCglobal Networks

Routing policy could be defined as a set of conditions specified by the network administrator. For example, a network administrator may decide to accept a path that has the smallest number of hops as a desired path, whereas others may decide to choose the path that has the lowest cost. The routing policy could be different from one network administrator to another; it depends on business agreements between the ASs [12].

Normally, when a node receives a message, it has to check the type of the message. If this message is a routing message, the node makes use of the configured policy to make decisions on whether to accept the advertised path or not. The following example illustrates an algorithm that prefers a path that has lowest number of hops over the cost. The example starts when the node receives the routing message; the node performs the following:

- It checks the AS_Path attribute. If this node finds that its AS name is in the received AS_Path attribute, it will reject this message [14], otherwise
- It will check its routing table for other routes to the same destination with a smaller number of hops. If there is one, then this new route is rejected, otherwise
- If there are two different routes that have the same number of hops, this node will check for the cost. If the cost of the received message is higher than the one stored locally, then this route is rejected, otherwise
- The received route overwrites the old route because it has the most recent status of the available EPC.

F. Advertising Policies

At this point, we are familiar with the needs of EPC-BGP routing protocol, we discussed earlier in this section the structure of our proposed protocol EPC-BGP. However, we did not explain anything about when are the routing messages exchanged, or what are the conditions that force an EPCglobal network to advertise?

We have proposed an advertising scheme based on the number of changes that took place in a given EPC-IS table. This can be done by introducing a counter at each EPC-IS table, which counts the number of changes of individual EPCs (being updated). When the value of the counter becomes higher than a threshold set by the network administrator, then an advertisement will be done.

If a node decides to advertise because a certain EPC-IS table experienced relatively high number of changes, hence that node will perform the following advertisement algorithm:

- For each row of column 5 in Table 1, the node intersects the available EPCs with the corresponding available EPCs in the EPC-IS table (Table 2).
- The advertising node inserts the result of EPCs intersection along with the corresponding route in an update message.
- the node modifies the originator, the cost and the As_path attributes and finally,
- The node sends the update message that includes the changes to its neighbors.
- The node update the EPC-IS table by resetting the status of the EPCs to TRUE once it sends the changes to other ECPglobal nodes.

G. Inter Domain Signaling

Up to this point, all we were talking about was the exchange of routing information among different ECPglobal networks; we defined the routing message types, mechanisms and specification of the proposed routing protocol.

Despite the fact that this work is about performing routing among different ECPglobal networks, nevertheless we decided to have a brief talk about a simple signaling protocol that will help us in evaluating our routing protocol in the context of our simulation studies. We used three types of messages for simulation purposes:

1. Request message
2. Confirm message
3. Tear down message

These three messages are used by each edge router to exchange new information about given EPC.

The common structure of the signaling message is in Figure 8.

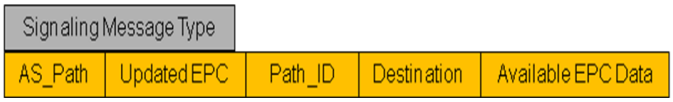

Figure 8: Signaling message structure

AS_Path: This attribute contains the list of the ASs that the signaling message has to traverse to reach the destination.
Updated EPC: This field specifies the EPC that has new data.
Path_Id: this is a unique number to identify the updated EPC.
Destination: It gives the destination EPCglobal network in which the new data has to reach.
Available EPC data: it holds the new data about a given EPC

H. Processing signaling messages among EPCglobal networks

Normally, the update Request Message will be triggered by the client application once new data is available about a given EPC, and will be sent after consulting the EPC-IS table for the status of that EPC. If the status is True, then the client application will change the status of the updated EPC in its EPC-IS table and send an update Request Message to the neighbor node that leads to the destination after increasing the counter that keeps track of the number of changes that occurred over its EPC-IS table. In turn, the neighbor node will check the destination field of the Request Message to find out if this update request is to be terminated at this EPC global network. If it does not recognize the destination field, the

neighbor node will get the new information about this EPC and then check the available resources for the updated EPC. If the requested EPC is available, it will change the status of this EPC and increase the counter at the EPC-IS table and finally forward the update Request Message to the next neighbor node that will lead eventually to the destination node. Once the destination node receives the update Request Message that has its address, it will send back a confirmation message back to the source node to confirm the update process has been completed.

Unfortunately, sometimes the requested EPC is not available along the entire path, it could be under a previous update process. Therefore, if an intermediate node finds that the requested EPC is no longer available, it will send back a tear down message to the source node indicating that this node has to try later.

I. Blocking Types

Unfortunately, it may happen that the EPC update could not be achieved when an update request is sent because the chosen EPC is under another Update request. This is what we called update blocking. For the purpose of this discussion, we assume that the information about all the network resources is correctly stored in a global table, i.e. if any EPC has been updated at a certain EPC-IS the global table is updated simultaneously with that EPC-IS.

In fact, there are three type of blocking that we are interested in:

- Justified Update Refusal. This blocking occurs when, both the global table and the routing table of the initiating node show that the requested EPC is already under a process of being updated.
  P_{JR}= Number of Justified Update Refusal/ Total number of requests
- Unjustified Update Acceptance. In this type of blocking, the global table shows that the requested EPC is not available for update, whereas the routing table of the edge node shows that the requested EPC is available. The reason for this difference is that each EPCglobal network may wait for a certain period of time to report about new updated EPCs to other EPCglobal network. Consequently, if the corresponding EPC at other EPCglobal network is being used, it will lead to blocking somewhere along the path and hence causing Unjustified Update Acceptance blocking.
  P_{UA}= Number of Unjustified Update Acceptance / Total number of requests
- Unjustified Update Refusal. In this type of blocking, the global table shows that the requested EPC is available for update whereas the distributed routing table of the EPCglobal network shows that the requested EPC is not available. Again, the reason for this difference is that each EPCglobal network may wait for a certain period of time to report about new released EPC to other nodes. Hence, other nodes assume that these resources are still reserved and hence never use them if a customer asks for them causing Unjustified Update Refusal blocking to occur.
  P_{UR}= Number of Unjustified Update Refusal / Total number of requests

Clearly, the total blocking probability is equal to the sum of three types of blocking.

$P_T = P_{JR} + P_{UA} + P_{UR}$.

The ideal case is to have both the P_{UA} and P_{UR} blocking equal to zero, this means that the distributed routing information is 100% correct and reflects accurately the topology and resource allocation in the global network. The above three mentioned blocking types are our major concern because they are directly related to the performance of our proposed routing protocol.

VI. Simulation Results

In our simulation model, we defined an EPCglobal object and an EPC-IS object. Each EPCglobal object represents an EPCglobal gateway that is run by a single administrator. Each node object includes the routing information stored in a table. The EPC-IS object has the information about the status of its EPC, both EPC-IS and the gateway are shown in figure 5.

Furthermore, we defined a global routing table; this table is used to keep track of the EPC status of all EPC in the simulated network model. This table has accurate information that can be used to verify the accuracy of the distributed routing information located in each node.

The global table and the information in the EPC-IS nodes are updated simultaneously each time a new update request is established or teared down. On the other hand, the routing tables of the gateways are only updated based on the advertising scheme explained earlier.

To evaluate the operation of the proposed protocol, we implemented several network architectures, namely the ARPANet architecture and the San Francisco architecture shown in Figure 9 and Figure 13, respectively. Finally, to ensure that our simulation results are accurate, we run each simulation for at least 10 times at each given rate of requests and calculated the average.

In our simulation, the requests are coming randomly and uniformly distributed over the nodes of the network. The requests arrive at the source node following Poisson distribution, and include the number of the requested EPC and the destination node. The selection of the destination node is also uniformly distributed over the total number of the network nodes.

There are many parameters for which we should investigate their effect on the performance of the routing protocol. These parameters are:

- Number of changes. This parameter represents the number of EPCs status changes in which it will trigger an advertisement.
- The number of EPC per EPC-IS.
- The request rate for new EPC update.

A. *ARPANet results*

Previously, we talked about the advertisement scheme that considers the number of changes that took place in the EPC-IS table to trigger an advertisement. This means that the advertisement is done based on the amount of information that has changed since the last up-date.

In this section, we will investigate the triggering of the advertisement based on the following percentage of change in the EPC-IS table: 5, 10, 15 %.

Figure 9 shows the ARPANet simulated network. We assume that each EPC-IS table has shipped 100 EPC to each destination.

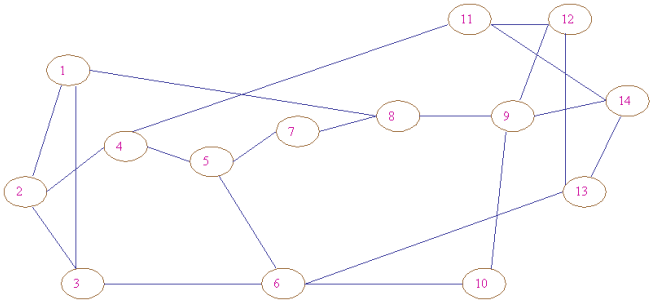

Figure 9: ARPANet.

Figure 10 shows P_{JR} at different thresholds. clearly Figure 10 shows that waiting for the counter to indicate five changes out of 100, i.e. "5% of change", does not have such a bad effect on P_{JR}, however, waiting for 15 changes out of 100, i.e. "15% of change", has a somehow worse effect on P_{JR}. This type of blocking indicates that the requested EPC is under a previous update and therefore, the update request will be blocked until the first update is complete. The reason for increasing $P_{JR\ as}$ as the threshold increased is that the longer the refreshment the longer the EPC status will be set to false and hence P_{JR} increased.

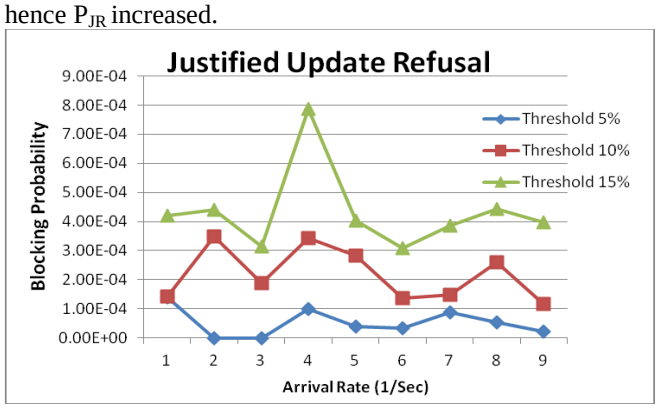

Figure 10: Justified Update Refusal

Figure 11 shows P_{UA} at different thresholds. The higher the threshold is the higher P_{UA} will be. P_{UA} is strongly affected by the threshold value. As we defined P_{UA} earlier, P_{UA} occurs due to slow reporting of the routing protocol. As a matter of fact, if the threshold value increases, this will cause lots of errors in the routing tables because each edge node will report changes slowly causing more errors in the routing tables and hence the requested EPC update will not be successfully done due to the fact that the initiation of requests based on wrong routing information will definitely lead to higher values of P_{UA}.

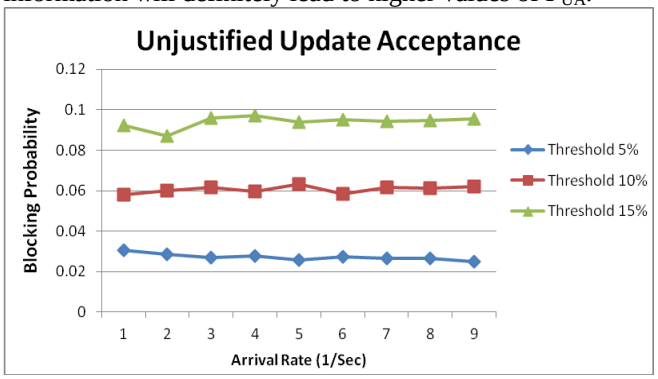

Figure 11: Unjustified Update Acceptance

Figure 12 shows that also P_{UR} is affected by the threshold value. It shows that P_{UR} is increasing as the threshold increased. Again, this type of blocking can be reduced by choosing low threshold as the routing tables will be updated more frequent and hence less errors in the EPC status

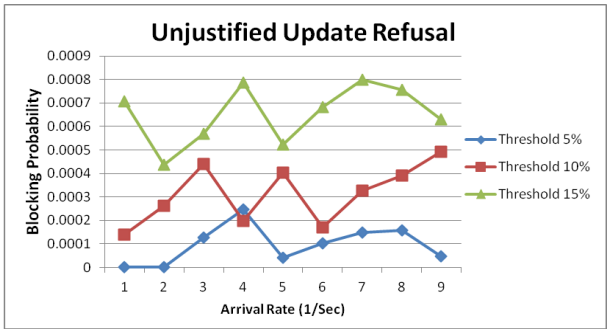

Figure 12: Unjustified Update Refusal

B. *San Francisco Network*

Figure 13 shows the San Francisco network, we have made the same assumption that each EPCglobal network has shipped a 100 EPC to each destination.

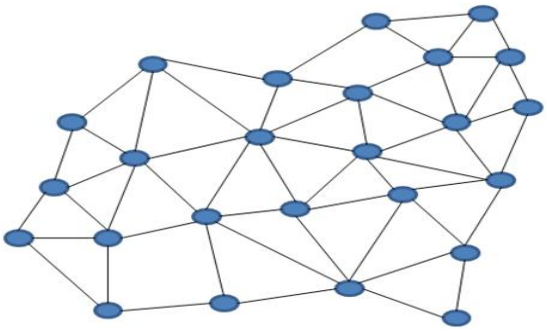

Figure 13: San Francisco Network

Figure 14, Figure 15 and Figure 16 shows the blocking probability for the EPC update table. It is clearly seen that the higher the threshold is the higher the blocking probability of the update is for the same reason explained in the previous section.

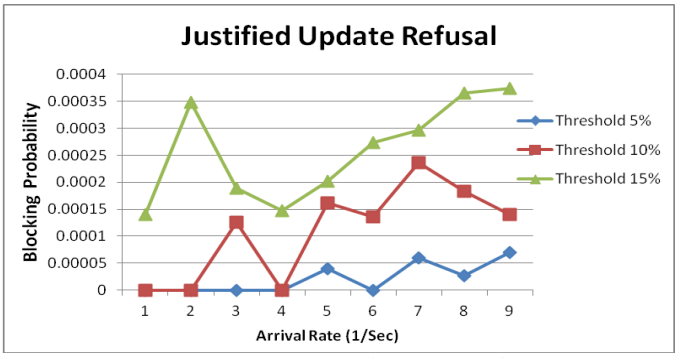

Figure 14: Justified Update Refusal

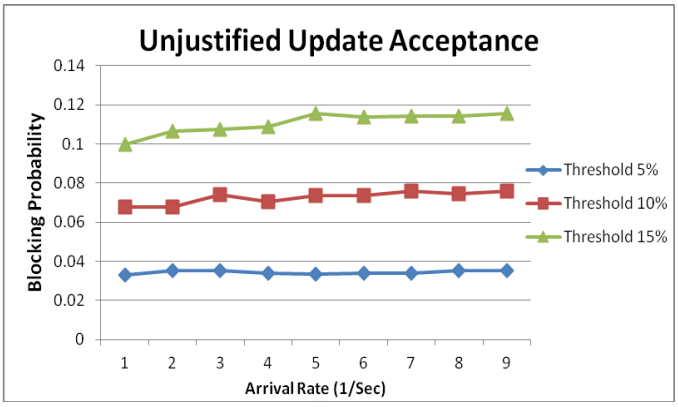

Figure 15: Unjustified Update Acceptance

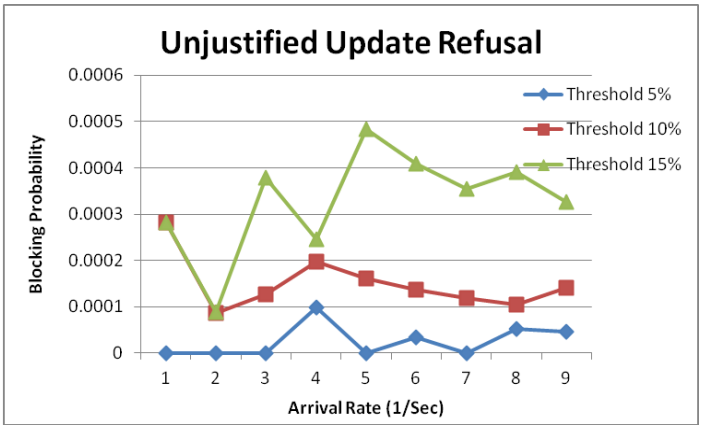

Figure 16: Unjustified Update Refusal

VII. Conclusion

Our major goal was to define a routing protocol that performs routing among EPCglobal networks to exchange information about objects in the supply chain. We proposed a new routing protocol called EPC-BGP. It exploits most of BGP functionality with some modifications to support the exchange of EPCs about shipped objects in the supply chain. We extended BGP to exchange the EPCs along with routing information by defining a new attribute called "EPC attribute". We defined a new advertising scheme based on the number of changes that took place in an EPCIS table. We investigate the performance of EPC-BGP on different network architectures and we showed that the blocking probability can be under control for small refreshing periods. We have seen that EPC-BGP relies on TCP connection for achieving secure connection between the communicated EPCglobal networks, furthermore, EPC-BGP allows policy configuration to prevent certain routes to be advertised to certain competitors.

References

[1] Ken Trau et al, "The EPCglobal Architecture Framework", EPCglobal Final Version 1.4 Approved 15 December 2010 http://www.gs1.org/gsmp/kc/epcglobal/architecture/architecture_1_4-framework-20101215.pdf

[2] S. Beier, T. Grandison, K. Kailing, and R. Rantzau. Discovery Services – Enabling RFID Traceability in EPCglobal Networks. In Proceedings of the 13th International Conference on Management of Data, 2006.

[3] D. Huang, M. Verma, A. Ramachandran, and Z. Zhou. A Distributed ePedigree Architecture. Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems, pages 220–230, 2007.

[4] EPCglobal, "EPCglobal Tag Data Standards Version 1.5," EPCglobal Ratified Standard,September 2011, ttp://www.gs1.org/gsmp/kc/epcglobal/tds/tds_1_6-RatifiedStd-20110922.pdf

[5] EPCglobal, "EPCglobal Object Naming Service (ONS), Version 1.0.1," EPCglobal Ratified Standard, May 2008, http://www.epcglobalinc.org/standards/ons/ons_1_0_1-standard-20080529.pdf.

[6] University of Cambridge, AT4 wireless, BT Research, and SAP Research. High Level Design for Discovery Services, 2007. BRIDGE project.

[7] EPCglobal, "EPC Information Services (EPCIS) Version 1.0.1 Specification," EPCglobal Ratified Standard, September 2007, http://www.epcglobalinc.org/standards/epcis/epcis_1_0_1-standard-20070921.pdf

[8] C. Kürschner, C. Condea, O. Kasten, and F. Thiesse. Discovery Service Design in the EPCglobal Network, pages 19–34. Springer Berlin / Heidelberg, 2008.

[9] M. A. Guijarro, G. Arrebola, J. J. Cantero, E. García, F. J. Nunez, J. Banos, M. Harrison, C. Condea, and H. Casalprim. Working Prototype of Serial-level Lookup Service, 2008. BRIDGE project.

[10] J. Muller, J. Oberst, S. Wehrmeyer, J. Witt, A. Zeier, H. Plattner, "An Aggregating Discovery Service for the EPCglobal Network", 43rd Hawaii International Conference on System Sciences (HICSS), Page(s): 1 – 9, 2010.

[11] D. Estrin, Y. Rekhter, S. Hotz "A Unified Approach to Inter-Domain Routing", May 1992, RFC 1322

[12] John T. Moy, "OSPF: Anatomy of an Internet Routing Protocol". ISBN 0-201-63472-4, Addison Wesley Longman, Inc., 1998.

[13] John W. Stewart, "BGP-4: Inter-Domain Routing in the Internet". ISBN 0-201-37951-1, Addison Wesley Longman, Inc., 1999.

[14] Y. Rekhter, "A Border Gateway Protocol 4 (BGP-4)", March 1995, RFC1771

[15] Stephen A. Thomas, "IP Switching and Routing Essentials: Understanding RIP, OSPF, BGP, MPLS, CR-LDP, and RSVP-TE". ISBN 0-471-03466-5, WILEY, Inc.,2001.

[16] Sergei Evdokimov, Benjamin Fabian, and Oliver Günther , "Multipolarity for the Object Naming Service ",In Proc. Of 1st International Conference on The Internet of Things- IOT'2008, LNCS 4952. Springer-Verlag, Berlin-Heidelberg, pp. 1-18, 208.

[17] Sandoche Balakrichenan, Antonio Kin-Foo, Mohsen Souissi , "Federated ONS Architecture for the Internet of Things - A Functional Evaluation", in Proc. Of Internet of Things 2010 Conference, Dec 2010.

[18] Olaf Maennel, Anja Feldmann, "Realistic BGP Traffic for Test Labs", In Proceedings of ACM SIGCOMM, pp. 235 - 247, August 2002.