A Circuit-based Approach for Routing in Unidirectional Links Networks

Thierry Ernst, Walid Dabbous

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

A Circuit-based Approach for Routing in Unidirectional Links Networks

Thierry Ernst & Walid Dabbous

N° 3292

October 1997

THÈME 1

ISSN 0249-6399

A Circuit-based Approach for Routing in Unidirectional Links Networks

Thierry Ernst & Walid Dabbous

Thème 1 — Réseaux et systèmes
Projet Rodeo

Rapport de recherche n° 3292 — October 1997 — 18 pages

Abstract: Current routing protocols assume that routers are connected by bidirectional links. However, in an increasing number of configurations, pairs of routers may only be connected by unidirectional links. Examples of such configurations include ad-hoc packet radio networks, some satellite and cable networks, etc. Specific protocols are needed to support dynamic routing over these configurations. In this report, we first discuss the applicability of current techniques to the support of dynamic routing in a network where all routers are connected by unidirectional links (UDLs). We show that these techniques cannot be used without modifications because of specific UDL characteristics. We then present an unidirectional link routing protocol based on circuit detection which allows dynamic routing over unidirectional links, and describes its features and behaviour.

Key-words: Circuit, Routing Protocols, Unidirectional link, Distance Vector, Link State, Acknowledgment, Flooding, Asymmetry

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 93 65 77 77 - International : +33 4 93 65 77 77 — Fax : 04 93 65 77 65 - International : +33 4 93 65 77 65
à partir du 01/01/1998
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Le Routage dans des Réseaux avec des Liens Unidirectionnels basé sur la découverte de Circuits

Résumé : Les protocoles de routage actuels se basent sur l'hypothèse que les routeurs sont connectés deux à deux par des liens bidirectionnels. Or, ceux-ci pourraient être connectés par des liens unidirectionnels dans un nombre grandissant de situations, comme par exemple des réseaux satellites, cablés ou radios. Ces configurations nécessitent des protocoles spécifiques pour supporter le routage dynamique. Ce rapport montre que les caractéristiques spécifiques aus liens unidirectionnels ne permettent pas d'utiliser les techniques usuelles de routage. Nous présentons donc un protocole de routage basé sur une nouvelle approche, la découverte de circuits, qui permet de faire du routage dynamique lorsque les liens sont unidirectionnels.

Mots-clés : Protocoles de Routage, Liens Unidirectionnels, Vecteurs de Distance, Etats de Liaisons, Inondation, Acquittements, Asymétrie, Tunnel

1 Introduction

Current routing techniques assume that pairs of routers are connected by bidirectional links (BDLs). UniDirectional Links (UDLs), i.e. links with zero return bandwidth, are nonetheless emerging. Unidirectional communication is experienced in several configurations namely 1) ad-hoc (wireless) networks where pairs of routers do not have the same transmitting power one to the other, 2) GEO satellite links with receive only hardware or 3) unidirectional cables.

Because current routing protocol assume that links are bidirectional, routers connected to an "outgoing" unidirectional link cannot be aware of the existence of networks reachable through this UDL.

Related work in this domain concerns the support of dynamic routing in the case of alternative unidirectional links on top of a bidirectional network i.e. a network where all links are BDLs. A solution based on tunneling is being discussed within the IETF `UDLR` working group [9].

However, this solution does not work in some cases where there are numerous UDLs. There is therefore a need for a routing protocol supporting such cases. We focus on the design of an intra-domain routing protocol for UDL networks i.e. networks where all links are unidirectional.

We discuss the applicability of current routing techniques such as Link State (LS), Distance Vectors (DV) and Path Vectors (PV) to support unidirectional link routing. There are limitations in using those techniques because of the bidirectionality assumption cited above. Moreover, some of the UDL characteristics require that a specific solution is designed rather than simply adapting current techniques. We propose such a solution, based on a protocol that detects *circuits*. Nodes located on those circuits are accessible nodes. We call our protocol "Circuit Discovery UniDirectional Link Routing Protocol".

The paper is structured as follows. In section 2, we present the solutions discussed within the `UDLR` WG, and we show why tunneling is not sufficient to support dynamic routing in all UDL networks. Section 3 details the impact of UDLs on routing techniques, whereas section 4 discusses around the applicability of known techniques to support UDL routing. Section 5 discusses the design issues and presents the details of a new protocol based on Circuit Detection. We conclude the paper in section 8, after discussing the advantages of our approach (in section 6) and the open issues (in section 7).

2 Tunneling: a solution for the simple case

A typical example of an UDL is a satellite link providing receive-only connectivity to the Internet with a terrestrial backchannel. This is an interesting case since satellites offer a high bandwidth transmission and a large geographical coverage. Their broadcast nature could be very useful for multicasting purposes. They can as well bypass bottlenecks by providing high bandwidth links over large geographical areas. Those advantages are of great interest to provide Internet access. However, the cost of VSAT antennas - the devices

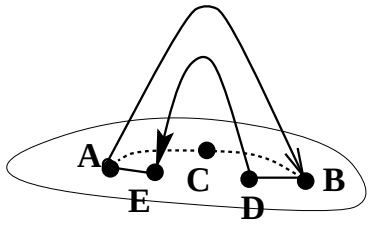

Figure 1: Receive-only connectivity via satellite

allowing emission of data toward satellites, is too prohibitive to be used on a large scale at the moment. The cost of receive-only hardware is nevertheless negligible. Satellites could therefore provide high speed one-way connectivity to the Internet. Reliable communication and dynamic routing require that a terrestrial backchannel exists in order to establish a bilateral communication. Tunneling allows to mask the underlying unidirectionality and to emulate a bidirectional link using the terrestrial bidirectional connectivity. As figure 1 shows, the terrestrial network provides the return path from B to A via C. Tunneling [8, 6, 7, 10] allows to give a quick solution without requiring any modification to existing routing protocols, neither to upper layers protocols and software. Another approach to solve this problem is to modify the routing protocols to take into account the undirectional link. Routing protocol modification [2, 4, 3, 1] is more efficient, it explicitly deals with the problem, but it requires changes to routing protocols implementations and may be in some other upper layers protocols which are highly dependent on the underlying structure (such as RSVP). Tunneling is therefore favoured for this specific case.

However, tunneling will not work in configurations where all links are unidirectional i.e. in an UDL network because tunneling assumes the existence of a bidirectional backchannel in order to establish the tunnel. This assumption is not longer true in all UDL networks. This could directly lead to deadlocks if the backchannel is itself unidirectional and therefore requires the setting of a tunnel. Figure 1 shows a situation where the backchannel from B to A via D and E is unidirectional.

We therefore focus on the design of a targetted solution, specifically adapted to UDL networks, that explicitly takes into account the underlying architecture of the network. It seems far better to treat the problem from scratch while still using the experience built from existing routing techniques.

3 Impact of UDLs on routing protocols

The obvious characteristic is that communication is not possible direcly in both ways. This gives rise to various considerations, like the cost of routing messages acknowledgments, how to deal with one-way reachable nodes, how to maintain the reachability information and how to deal with the assumption that metrics are symmetric.

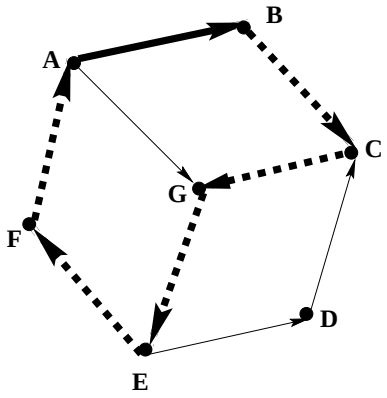

Figure 2: Example UDL network

3.1 Hop-by-hop acknowledgments adds overhead

Many routing protocols require that routing messages are acknowledged on each link i.e. by the direct neighbour. In BDL networks, these "hop-by-hop" acknowledgments are sent on the same link than the routing messages themselves. In an UDL network, those acknowledgments would have to follow a (sometimes much) longer path than that taken by the routing messages, this leading to overhead.

Figure 2 shows an example UDL network. The acknowledgements of routing messages sent from A to B will travel from B to A through $BCGEFA$ compared to the direct BA link if this was a BDL network. It is therefore useful to use a protocol that does not require routing messages to be "hop-by-hop" acknowledged.

3.2 One-way connectivity is not enough

Providing one-way connectivity only is not enough since most applications require a bi-directional communication either to maintain interactivity or to ensure the transmission reliability. Each node should therefore be able to send and receive data to every other node in the network.

In a BDL network it is sufficient that the graph is "connected" (i.e. every pair of nodes is connected by a link) in order to be able to ensure bidirectional communication between all nodes.

On the other hand, it is not sufficient in an UDL network that the "undirected version" of the graph is connected in order for the directed graph to be strongly connected (i.e every two nodes are reachable from each other). In other words, each node require at least one "predecessor" and one "successor".

That means that the routing protocol should consider a node unreachable if there exists no incoming link or if there exists no outgoing link. For example, if the link FA is down in

the example UDL network drawn in Figure 2, nodes F, A and B should be considered as unreachable.

3.3 Reachability information is harder to maintain

Routing protocols include mechanisms to provide the knowledge of direct neighbours and ensure their reachability, in particular after a failure or a recovery.

In a BDL network, listening to periodic messages from neighbours is sufficient in order to determine easily which are the direct neighbours whereas in an UDL network only predecessors can be determined using this mechanism. There is no such straight mechanism to provide knowledge of successors. This requires an additional mechanism to explicitly advertise successors back to their predecessors.

Neighbourhood reachability is therefore harder to ensure in an UDL network since there is no direct backchannel to provide the identity and the operationality of successors back to the local node. However, by monitoring the (non) reception of periodic messages, successors are aware in case of failure or recovery of their predecessors. They should have the responsibility to inform the predecessors of the failing link or node. A node should therefore consider its successors as operational until the contrary is explicitly specified.

3.4 Metrics are not symmetric

Many routing protocols assume that links metrics are symmetric. Based on this assumption, routers deduce reverse routing metrics information from routing messages they receive. This is no longer true in an UDL network. As shown in figure 2, the shortest distance from A to B (=1) is not equal anymore to the shortest distance from B to A (=5).

Obtaining routing information is therefore more expensive in an UDL networks than in BDL ones as one could expect. An adequate mechanism that does not make the symmetry assumption is therefore needed in order to be able to build the routing tables.

4 Applicability of known techniques

In this section, we study the techniques used by the current routing protocols in the Internet, trying to point out their applicability and limitations within an UDL network.

The techniques used by current routing protocols essentially are Link State (LS) and Distance Vectors (DV) which are used in IGPs (intra-domain protocols), and Path Vectors (PV) which is currently used by BGP, the most popular EGP (inter-domain protocol).

4.1 Link State

The LS technique can be summarized in a few words: each router determines its own connectivity which is thereafter flooded to the entire network. LS protocols such as OSPF, are seen in the literature [5] as very powerful protocols since routes are computed at each node with

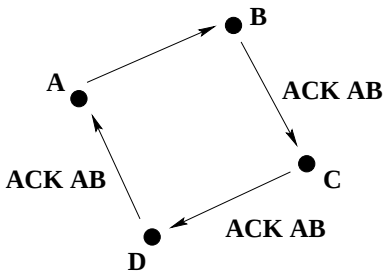

Figure 3: Link State Acknowledgement concerning link AB

a total knowledge of the topology, based on (1) the exchange of databases between adjacent routers and (2) the flooding of Link State Updates. This is necessary to provide synchronized full topology knowledge to all nodes. However, these two operations are "acknowledged". In the Database exchange process, Database Description packets sent by one of the routers ("the master") are acknowledged by the "slave" with responses containing a summary of its link state data. As for Link State Updates (LSUs), they are acknowledged by sending Link State Acknowledgment packets back to the sending neighbour.

As described in section 3.1, these "hop-by-hop" acknowledgements consume more bandwidth in an UDL network. This is shown in the example in figure 3. Providing the knowledge of link AB to the network require 4 LSUs to be sent, and each of the 4 corresponding acknowledgments would travel on 3 links.

The Acknowledgement of LSU packet sent from A to B will be sent on three links (BC, CD and DA) in this UDL network instead of being sent back on the same link if the network was BDL.

More importantly nodes don't know yet how to forward the acknowledgement back to A at the time of this topology discovery. This brings serious limitations to the use of LS. We conclude that the acknowledgement of routing traffic on each link is an important drawback of the LS technique.

4.2 Distance Vector

DV protocols are very simple and easy to implement, but provide less facilities than LS. Although one could argue it is possible to bring the same facilities up to DV protocols, the necessary addition of complexity would nevertheless highly diminish the advantage of DV over LS.

Even without considering UDL networks, DV already exhibit one important drawback. In order to avoid permanent loops, a mechanism called "counting to infinity" is binding the size of the network to a small limit in term of hop count, not to say that only one metric is allowed.

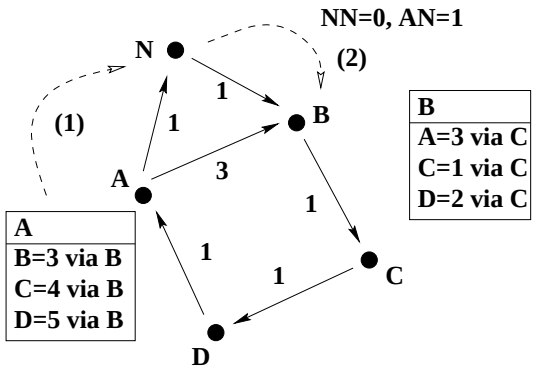

Figure 4: DV is not adequate for UDL networks

In the DV technique, a router E receiving that R can reach X with metric n deduces that E itself can reach X with metric $n+m$, where m is the metric of the RE link (in both directions. In turn, E broadcast its DV to its neighbours.

In UDL networks, nodes cannot conclude their own distance to destinations from the DV obtained from their predecessor. The routing information obtained by this technique is therefore useless if the technique is not adapted. Figure 4 gives an example. A new node N cannot do anything with the information contained in the DV received from A. It only concludes the distance from A to N.

It is however possible to adapt the DV technique for UDL networks. In fact, N knows the distance from A to N and can forward this to B. B cannot deduce anything directly from this information, but it forwards not only ($NB = 1$ via B) and ($AB = 2$ via N), but also ($AN = 1$ via N) so that eventually A can deduce its distance to get to N and B, as shown in figure 5.

In addition, flooding is necessary since routers don't already have the knowledge of routes to get to the initiating router. We therefore see that adapting DV adds overhead since all distances between any two routers are as well flooded.

The important difference between UDL and BDL networks, is that, in an BDL, protocols start the discovery of destinations based on the knowledge of direct neighbours. In an UDL, this knowledge is easy to obtain from the predecessor to the successor but not in the other direction. Moreover, because routes are not yet discovered, the only possible technique to transmit this information is flooding.

4.3 Path Vectors

The concept of "Paths" makes the study of PV interesting. In PV, loops are avoided by advertising the complete path to each destination instead of counting to infinity. Paths are therefore not limited to the infinity value and various metrics can be used. Each router only advertises paths it itself uses similarly to DV protocols. Upon reception of a new path, a

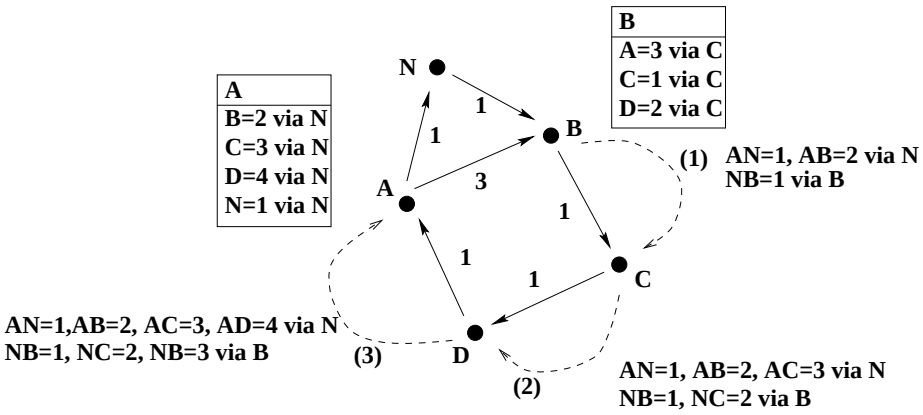

Figure 5: DV adapted to UDL networks is complex

node checks whether it is shorter than what it has in his table and only advertises shorter paths.

However, PV faces the problem of paths asymmetry in UDL networks. For example, a router E receiving that A can reach E via nodes B, C and D with metric m, deduces that E can reach A with the same metric m. The PV technique is therefore not usable without adapting it to UDL networks.

5 A Circuit Discovery Unidirectional Link Routing Protocol

We have seen in section 3.2 that a bidirectional communication is always required. In a UDL network, this means that the local node can reach a particular destination if it finds a *circuit* that goes from the local node to the destination and comes back, likely by a different path. By definition, a circuit is a sequence of unidirectional links getting to a particular destination and coming back to the departure node such as $AGEFA$ in figure 6 (a).

Determining the best path to get somewhere is therefore equivalent to find the best circuit, which is as well the aggregation of the best path from the source to the destination and the best path from the destination to the source.

Detecting circuits could therefore inform the local node of the reachable nodes, which are all nodes located on circuits since there is a path getting to the destination and one coming back to the source.

The return path might sometimes take a common part of the outward path, depending on the structure of the network. This is shown in figure 6 (b) where a circuit originated at D and aiming to reach B will have to pass two times on links CG and GE. This means that a subset of nodes of the circuit forms itself a circuit, like $GEFABCG$ in figure 6 (b).

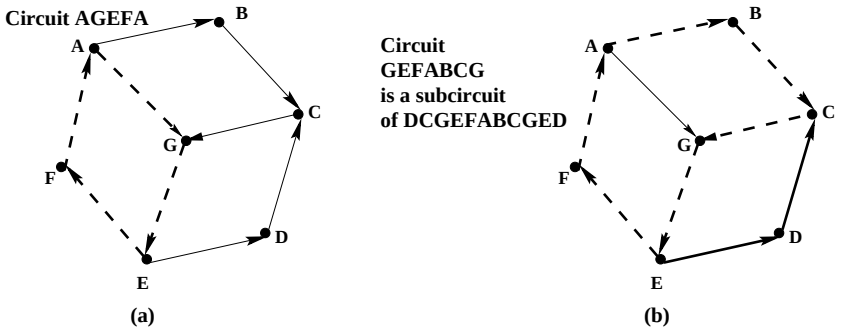

Figure 6: Circuits and Subcircuits

We will call this a *subcircuit* of circuit $DCGEFABCGED$. A circuit is allowed to use a subcircuit only if the destination is located on it. This limits the number of subcircuits to one.

We propose a dynamic routing protocol for UDL networks based on the circuit concept. We list hereafter some of the design issues, and then we describe the protocol.

5.1 Design Issues

The main requirement is that our protocol should allow routers to take routing decisions in an UDL network. In short, based on topology information flooding, the routers should be able to take correct routing decisions just as they would do in a BDL network.

We also want this protocol to provide multipath routing, various metrics support, quick adaptation in the face of a failure, to be scalable for large routing domains.

Moreover, the routing traffic should be kept low if possible. It should therefore avoid overhead and use the available information in the most efficient way. The computational capacity is of less importance at this stage.

5.2 Description of the protocol

Our protocol finds circuits from which it deduces possible destinations. Because consistency is limited from circuit to circuit, routing of data packets can start as soon as a circuit is found and the scope of circuits can be limited in order to adapt to the size of the network. It might be necessary in large networks to avoid an increasing length and high number of circuits. The protocol therefore doesn't require the full knowledge of the topology.

The protocol is based on the detection of circuits. All nodes on a circuit are potential destinations and we know we can establish a communication in both directions with all nodes located on the circuit once we have found a path going back to the departure place.

The aim of the protocol is to therefore to maintain tables containing at each node of the selected circuits. The protocol doesn't necessarily compute circuits for each destination neither select all possible circuits for a particular destination.

The protocol is based on "one-way" *Link Advertisement Messages* (LAMs) sent by routers in order to announce *new* links to all their successors. Each node has a "local topology graph" built with the information received in LAMs. When a router receives a LAM containing new information it updates its local topology graph and sends a LAM including only the new information on all outgoing links i.e. to all successors. If the received LAM does not modify the topology graph no LAM is sent out.

However, all nodes send periodic *KeepAlive Messages* (KAMs) to their successors. Successors are therefore aware of the failure of their predecessors or of the link from them, if they don't receive any KAM during a given time period.

A *decision process* running at each node has the responsibility to compute the "possible circuits" based on the local topology graph. A part of these circuits is selected and stored in a "Circuit Table" labeled as "selected circuits". A mechanism to validate these circuits is then launched. If this mechanism succeeds, the circuit is labeled as "enabled".

New nodes are noticed by their successors which in their turn, inform the network about the existence of a new link using LAMs. Upon failure, the immediate successors have the responsibility to inform all nodes which appear on circuits involved by the failure using a *Link Update Message* (LUM). Those circuits are "disabled" until the failure recovery, and a recovery timer is set. When a link recovers from a failure, the circuits involved are enabled again. If the recovery timer has expired, the corresponding circuits will be deleted from the table. If the link recovery takes place after the expiry of the timer, it will be processed in the same way as the addition of a new node.

Here follows a more detailed description of the different protocol elements.

The Link Advertisement Message The LAM is sent by every node at pseudo regular intervals (in order to limit the number of LAMs), only if the local topology graph has been updated since the last LAM. It is forwarded by the node to all its successors after adding the information regarding the "incoming" links (including e.g. the link metrics) to the local topology graph, if it's not already recorded. Therefore, only the new information added in the tree is forwarded to its sucessors in order to avoid redundancy since the information already in the table has already been forwarded.

The Circuit Table Each node maintains a table containing the computed circuits. These circuits are labeled as "possible" circuits. However, it may not be interesting to validate all these circuits for different reasons (e.g. several circuits leading to the same direction). The "selected" circuits are the ones we choose to validate. If the validation process succeeds, the circuit is enabled. Otherwise it is disabled and possibly deleted, as described by figure 7. Possible destinations are the nodes contained in the possible circuits. The routing table is built based on the enabled circuits information. The shortest circuit containing a given

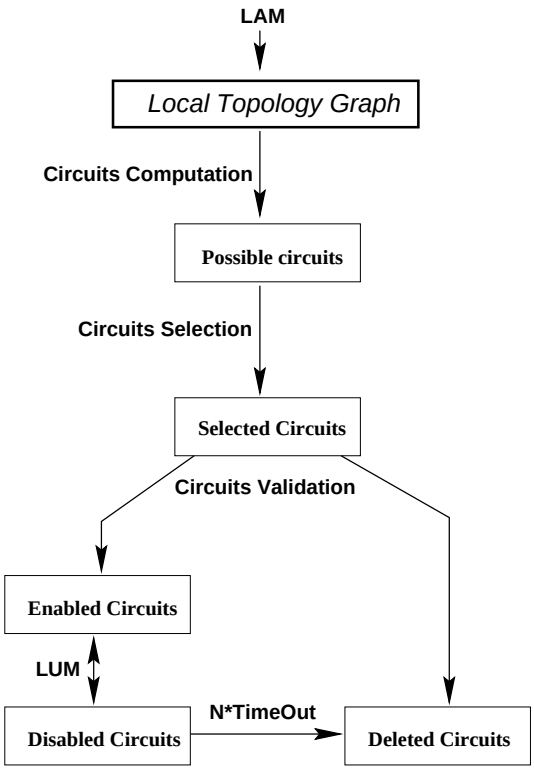

Figure 7: Circuit selection and validation

destination includes both the shortest path to go to this destination and the shortest to come back.

Circuit Selection by the Decision Process The decision process computes circuits from the tree and records them in the Circuit Table. It might take different decisions in each node concerning which part to select from the possible circuits. Basically, we could limit the knowledge of circuits in a nearby area. On the other hand, keeping several circuits for a particular destination allows to provide quickly a substitution circuit in the face of failure. It as well allows multipath routing. The decision process computes a degree of preference for each possible circuit and labels the prefered ones as selected. A selected circuit must first be validated by all nodes which appear on it before being enabled. The selection of circuits is implementation-specific and is therefore outside the scope of this paper.

Validation of Selected Circuits Before being used for routing, circuits selected by the decision process have to be validated amongst the nodes which appear on them. All the nodes of a given circuit should be aware of the circuit in order for it to be enabled. Once a circuit is selected, a *Circuit Validation Message* (CVM) is sent by the node with the smallest network address (called "Circuit Origin") in order to validate the circuit. The CVM contains the network addresses of the circuit nodes, an iteration number and a time-out value t_{val} based on the circuit length. The Circuit Origin also sets a retransmission timer to t_{val}. Each node forwards the CVM to the successor on the circuit, and sets a validation timer to t_{val}. When the CVM comes back to the Circuit Origin, it declares the circuit as enabled. If the message doesn't come back after t_{val}, the CVM is retransmitted, with the iteration number increased by one. As for intermediate nodes, they declare the circuit enabled after the validation timer expiry unless they receive a failure message (see paragraph failure hereafter) or a validation message with an increased iteration number. Consistency is limited from circuit to circuit and is ensured when the CVM comes back to the Circuit Origin. It guarantees that all nodes that appear on the circuit have its knowledge.

KeepAlive Messages The identity and the operationality of predecessors is ensured by KAMs sent on each "outgoing" link. Information concerning the successors is provided by the information recorded in the Circuit Table. Successors are considered operational unless their failure is explicitly notified. Upon a topology change, the immediate successors are responsible for the proper delivery of notification to the interested nodes of the network. KAMs are sent at pseudo regular intervals to avoid the synchronisation of the routing messages. However, KAMs are sent only if no routing messages are sent for more than a given period.

Addition of a new link or node KAMs are immediately sent on new links, so that successors are aware of their new predecessor. Successors have the responsibility to inform the rest of the network (and therefore the predecessor) that the link is operational. From this point, the new portion of path is advertised to all successors using LAM. As an instance

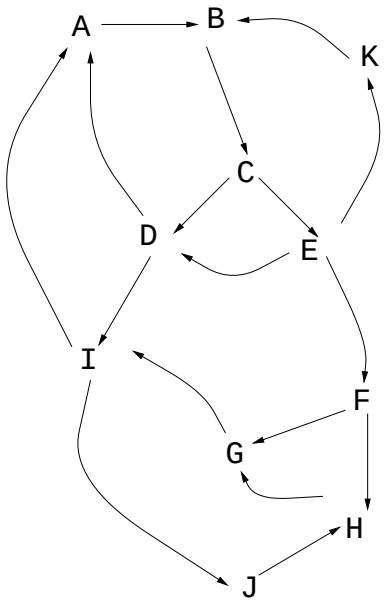

Figure 8: Demonstration Network

in our figure 8, the addition of the link DA in the network would be notified to A by KAMs received directly from D. A sends a LAM advertising the new portion DA, so that D for instance would deduce other circuits to reach A. If it already knows the circuit $DIABCD$, it would deduce $DABCD$. In the case of addition of a new node, a LAM is sent to the successors. When the predecessors of the new node receive the LAM containing its description, they send all their local topology graph to it.

The Link Update Message Upon detection of a failure, the immediate successors notice it and forward this information to all nodes contained in the circuits involved using a LUM. Each node marks relevant circuits as disabled and sets a timer. Its expiry guarantees that circuits are permanently deleted from the table if links don't recover from their failure after a specified amount of time. In the case of failure of link DA in figure 8, nodes I, J, H, G and F might not have validated a circuit passing via this link. Circuits involved are therefore $ABCDA$, $ABCEDA$, $ABCEKBCDA$ (used to reach K from A). Node D doesn't forward the LUM to I since it notices there are no circuits passing through DA and I. Once links have recovered from their failure, the immediate successors receive again KAMs. The information is sent to all nodes contained in the circuits involved which set relevant circuits in the state enabled again. The recovery is equivalent to the addition of a new link if entries were deleted because the timer has expired.

5.3 Large Networks

In significantly large or dynamic networks, it would be inappropriate to compute circuits to reach all destinations in the network. This would require too much time and bandwidth.

Some protocols address this problem by doing on demand routing, based on the fact that most traffic is usually limited in the nearby area anyway. The cost of on demand routing, that is, the higher delay to find a valid route, can well be balanced to the cost of establishing a full knowledge of the topology.

In our protocol, the decision process can select circuits according to their length, e.g. whose length is inferior to a given threshold. Each node therefore records a distinct set of circuits. Some nodes might have an extended view of the network.

On demand routing doesn't require the full knowledge of the topology. Since consistency is limited from circuit to circuit, this help to decrease as well the cost in the face of topology changes. Only circuits passing through the link involved in the change have to be updated.

5.4 Simulation of the protocol

We have simulated the protocol in order to verify its correct behaviour. The simulated prototype emulates the routing traffic that takes place between a small number of nodes. We have applied this simulator to the demonstration network shown in figure 8. It performs well and effectively provides to all nodes the knowledge of links via LAMs. The decision process computes all possible circuits.

We based the selection of circuits on hop count. The prototype therefore found the shortest circuit allowing to reach each node. The validation is initiated by the node with the smallest number (Circuit Origin) and each node that appear on the circuit correctly gets and forwards the CVM.

In the demonstration network, there are 17 links and 11 nodes. In total, 136 LAMs and 54 CVMs are sent on the different links. The circuits validated are shown in table 1. Their lengths totalize 54 links, which is the number of CVMs sent.

Providing the topology to all nodes using the LS technique would require to send link advertisement messages on $17^2 = 289$ links, without considering the messages required for the acknowledgements.

6 Advantages of the approach

Although this protocol is still undergoing studies, we can already exhibit some of its advantages.

First, our approach allows the adaptation to failures by quickly informing the network about such failure without directly deleting the circuits involved, while also providing for multipath routing.

Second, there is no need for explicitly maintaining globally consistent tables. Routing is done accurately if all nodes that appear on a circuit have the same vision of this circuit.

	Circuit	Length	Circuit Origin
1	A B C D I A	5	A
2	A B C D A	4	A
3	A B C D I J H G I A	9	A
4	A B C E F H G I A	8	A
5	A B C E F G I A	7	A
6	A B C E D A	5	A
7	A B C E K B C D A	8	A
8	B C E K B	4	B
9	G I J H G	4	G

Table 1: Validated Circuit

The consistency is therefore limited from circuit to circuit. It doesn't prevent our protocol to work with suboptimal circuits in the case of transitory periods.

Third, our protocol allow to perform routing even in the case of partial knowledge of the topology. All destinations that appear on enabled circuits are reachable even if all possible circuits are not validated yet. This is cheaper than flooding to the entire network. Although similar to the LS technique, our technique requires less messages since this information is not sent to the entire network and doesn't require hop-by-hop acknowledgements.

Fourth, this protocol may also be used to support routing over asymmetric links. In fact, some links have an asymmetric bandwidth and many applications have asymmetric data traffic (e.g. ftp or web traffic). Current protocols in use set the same metric in the two directions. This asymmetry is therefore not taken into account for routing decisions. Splitting off one BDL in two UDLs allows our protocol to support routing over asymmetric links.

7 Open Issues

There are numerous open issues currently over investigation that could enhance our protocol.

First, the protocol doesn't take into account point to multipoint unidirectional links. Basically, this could deserve importance, since this is the case for satellite links. The proposed protocol deals only with point to point unidirectional links. Further study should therefore address the case of multicast unidirectional links.

We have focused on networks where all links are UDLs. Our protocol seems well adapted under the assumption that the return path is taking a route highly different than the one taken for the outward path, i.e. if an important portion of the links are not actually bidirectional. In the case of a network where a large part of the links are BDL, a specific mechanism to detect such links is required. In our protocol, this would require to detect nodes

sequences such like XYX, i.e. circuits involving only two nodes. An interesting question is how to allow the cohabitaion of both BDLs and UDLs in an optimum manner.

Amongst important topics under investigation is the decision process. Various reasons might decide for the non selection of circuits. We have outlined some of the "heuristics" used in the previous sections. Particularly, we are considering nodes or portions of circuit that are *inevitable* gateways in order to reach some destinations, such as node E if we want to reach K in figure 8. Nodes willing to access destinations beyond inevitable nodes don't require the knowledge of the path. It will be the responsibility of the inevitable node to correctly route packets beyond this point. An interesting enhancement is therefore to dynamically discover those inevitable nodes in order to reduce the routing table size.

8 Conclusion

In this paper, we have shown the need for a new protocol to support dynamic routing in the case of all UDLs networks. We studied the current techniques in use in the Internet, namely Link State, Distance Vectors, and Path Vectors. We proposed a protocol based on the discovery of "circuits" that does not require hop-by-hop acknowledgment of routing messages. It ensures bidirectional communication and provides reachability information.

Our protocol called *Circuit Discovery Unidirectional Link Routing Protocol* provides for multipath routing, and doesn't require a full knowledge of the topology. It doesn't necessarily provide all circuits, those might even been computed on demand only.

References

[1] Emmanuel Duros and Walid Dabbous. Handling of Unidirectional Links with DVMRP. Internet draft <draft-ieft-dvmrp-unidirectional-link-00.txt>, INRIA Sophia-Antipolis, March 1996. available at http://www.inria.fr/rodeo/udlr.

[2] Emmanuel Duros and Walid Dabbous. Supporting Unidirectional Links in the Internet. In *Proceedings of the 1st International workshop on Satellite-based Services*, Rye, New-York, November 1996. INRIA Sophia-Antipolis, France.

[3] Emmanuel Duros and Christian Huitema. Handling of Unidirectional Links with OSPF. Internet draft <draft-ieft-ospf-unidirectional-link-00.txt>, INRIA Sophia-Antipolis, March 1996. available at http://www.inria.fr/rodeo/udlr.

[4] Emmanuel Duros and Christian Huitema. Handling of Unidirectional Links with RIP. Internet draft <draft-ieft-rip-unidirectional-link-00.txt>, INRIA Sophia-Antipolis, March 1996. available at http://www.inria.fr/rodeo/udlr.

[5] Christian Huitema. *Routing in the Internet.* Prentice Hall, 1995.

[6] Hidetaka Izumiyama and Akihiro Tosaka. Uni-directional Link Routing with IP tunneling. Internet Draft <draft-ietf-wide-udlr-vif-00.txt>, Keio University, WIDE Project, April 1997. available at ftp://zenon.inria.fr/rodeo/udlr/archive.txt.

[7] Venkata Padmanabhan, Hari Balakrishnan, Keith Sklower, Elan Amir, and Randy H. Katz. Networking using Direct Broadcast Satellite. In *Proc. of the 1st international Workshop on Satellite-based Services*, Rye, New-york, November 1996. University of California at Berkeley.

[8] James Stepanek and Stephen Schwab. Virtual internet Packet Relay - An Encapsulation Architecture for Unidirectional Link. Internet draft <draft-stepanek-vipre-00.txt>, The Aerospace Corporation and Twin Sun Inc, March 1997. available at ftp://zenon.inria.fr/rodeo/udlr/archive.txt.

[9] UDLR web page. http://www.inria.fr/rodeo/udlr.

[10] Yongguang Zhang and Son Dao. Integrating Direct Broadcast Satellite with Wireless Local Access. In *Proc. of the 1st international Workshop on Satellite-based Services*, Rye, New-york, November 1996. Hugues Research Laboratories.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399