Virtual Ring Routing

Amalrose Vayalinkal

University of Toronto

August 2021

Table of Contents

The Set-Up

Multi-hop routing algorithms describes a scheme for communication between devices.

The Set-Up

Multi-hop routing algorithms describes a scheme for communication between devices.

Main Goals for a Multi-hop Routing Scheme:

- Every device can communicate with any other device
- Want to forward messages efficiently

The Set-Up

Multi-hop routing algorithms describes a scheme for communication between devices.

Main Goals for a Multi-hop Routing Scheme:

- Every device can communicate with any other device
- Want to forward messages efficiently

Main Concerns When Routing:

- Memory capacity constraints
- Over-reliance on specific nodes in the network
- Scalability
- Length of routing path

The Premise of VRR

Utilizing two different networks, the physical network and the virtual network, VRR determines routing paths between nodes independent of their location.

The Premise of VRR

Utilizing two different networks, the physical network and the virtual network, VRR determines routing paths between nodes independent of their location.

VRR schemes can be modelled by a pair of graphs, the physical network and the virtual network.

The Premise of VRR

Utilizing two different networks, the physical network and the virtual network, VRR determines routing paths between nodes independent of their location.

VRR schemes can be modelled by a pair of graphs, the physical network and the virtual network.

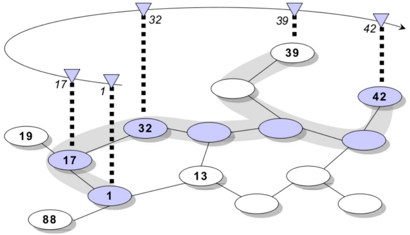


Figure: The underlying physical network and the virtual overlay (image from Wikipedia)

The Physical Network

An undirected graph $G = (V; E)$

The Physical Network

An undirected graph $G = (V; E)$

- Vertices in V are called nodes, $jVj = N$

The Physical Network

An undirected graph $G = (V; E)$

- Vertices in V are called nodes, $jVj = N$
- Edges denote physical links

The Physical Network

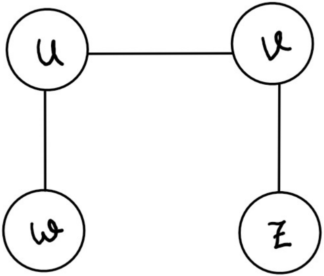
An undirected graph $G = (V;E)$

- Vertices in V are called nodes, $jVj = N$
- Edges denote physical links
- G must be connected

The Physical Network

An undirected graph $G = (V; E)$

- Vertices in V are called nodes, $jVj = N$
- Edges denote physical links
- G must be connected



$(u; v)\ 2\ E$ is a physical link

The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

- Virtual links are formed by chaining physical links (and/or preexisting virtual links) that share an endpoint.

The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

- Virtual links are formed by chaining physical links (and/or preexisting virtual links) that share an endpoint.
- We allow chains of length 1, so physical links can also be considered virtual links

The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

- Virtual links are formed by chaining physical links (and/or preexisting virtual links) that share an endpoint.
- We allow chains of length 1, so physical links can also be considered virtual links

The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

- Virtual links are formed by chaining physical links (and/or preexisting virtual links) that share an endpoint.
- We allow chains of length 1, so physical links can also be considered virtual links

If $hu; vi\ 2\ E_v$, then $hv; ui\ 2\ E_v$.

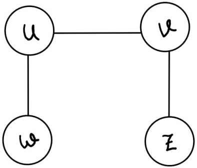
The Virtual Network

Definition 1

Given a physical network $G = (V; E)$, we can construct a **virtual network** $G_v = (V; E_v)$, where E_v denotes **virtual links**.

- Virtual links are formed by chaining physical links (and/or preexisting virtual links) that share an endpoint.
- We allow chains of length 1, so physical links can also be considered virtual links

If $hu; vi\ 2\ E_v$, then $hv; ui\ 2\ E_v$.



$hu; zi\ 2\ E_v$ is a virtual link, where $hu; zi = vl((u; v); (v; z))$

Identifier Space

To set up routing, we need an identifier space Ω:

- well-ordered set
- distance function d

Identifier Space

To set up routing, we need an identifier space Ω:

- well-ordered set
- distance function d

Also need a function that assigns each node to a unique $id\ 2\ \Omega$

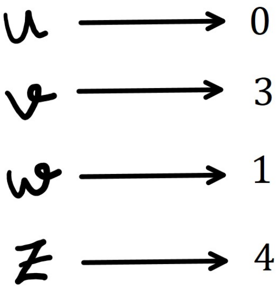
Identifier Space

To set up routing, we need an identifier space Ω:

- well-ordered set
- distance function *d*

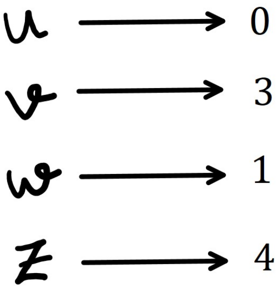
Also need a function that assigns each node to a unique *id 2* Ω
In VRR, we always take Ω N and assume the function assigns *id* randomly

Node to Id Mapping

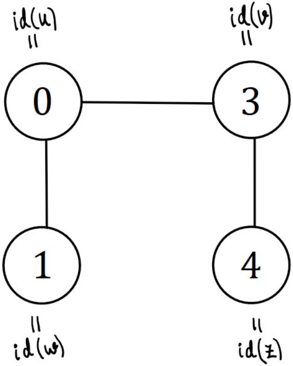


Node to Id Mapping

Node to Id Mapping



Node to Id Mapping



We often use v or $id(v)$ to refer to the same node

Table of Contents

Constructing the Ring

For each node $v \in V$, we set up a virtual link between v and the node with the next largest identifier v_+.

Constructing the Ring

For each node $v \in V$, we set up a virtual link between v and the node with the next largest identifier v_+.
i.e. $\langle v, v_+ \rangle \in E_v$

$$id(v_+) = \min\{id(x) \mid x \in V, id(x) > id(v)\}$$

Constructing the Ring

For each node $v \in V$, we set up a virtual link between v and the node with the next largest identifier v_+.
i.e. $\langle v, v_+ \rangle \in E_v$

$$id(v_+) = \min\{id(x) \mid x \in V, id(x) > id(v)\}$$

If $id(v)$ is the largest, then v_+ is the node in V with the smallest id

Constructing the Ring

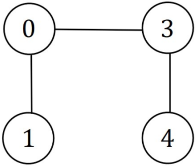
For each node $v \in V$, we set up a virtual link between v and the node with the next largest identifier v_+.
i.e. $\langle v; v_+ \rangle \in E_v$

$$id(v_+) = \min\{id(x) \mid x \in V; id(x) > id(v)\}$$

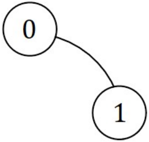
If $id(v)$ is the largest, then v_+ is the node in V with the smallest id

Note: It is always possible to create a virtual link between any two nodes in V since we require G to be connected

For Example...

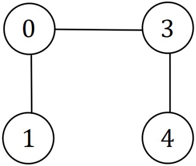


Physical Network

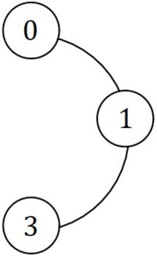


$\langle 0; 1 \rangle = vl(0; 1)$

For Example...



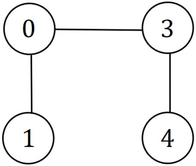
Physical Network



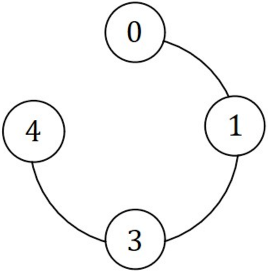
$$h0; 1i = vl(0; 1)$$

$$h1; 3i = vl((1; 0); (0; 3))$$

For Example...

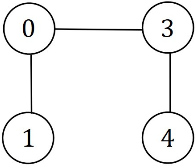


Physical Network

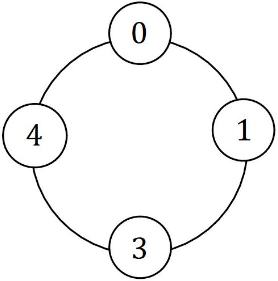


$$\langle 0; 1 \rangle = vl(0; 1)$$
$$\langle 1; 3 \rangle = vl((1; 0); (0; 3))$$
$$\langle 3; 4 \rangle = vl(3; 4)$$

For Example...



Physical Network

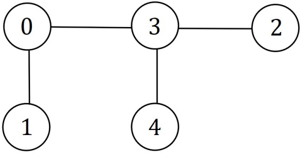


Virtual Network

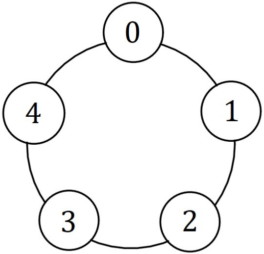
$$h0;1i = vl(0;1)$$
$$h1;3i = vl((1;0);(0;3))$$
$$h3;4i = vl(3;4)$$
$$h4;0i = vl((4;3);(3;0))$$

Another Node?

If we add another node then



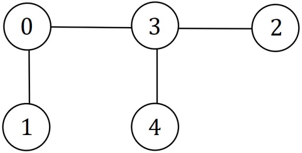
Physical Network



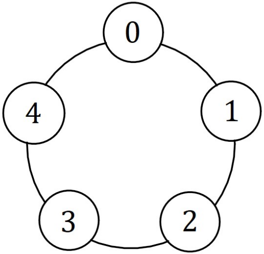
Virtual Network

Another Node?

If we add another node then



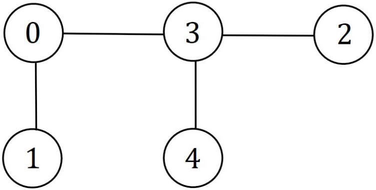
Physical Network



Virtual Network

$$h1;2i = vl(h1;3i;(3;2))$$
$$h2;3i = vl(2;3)$$

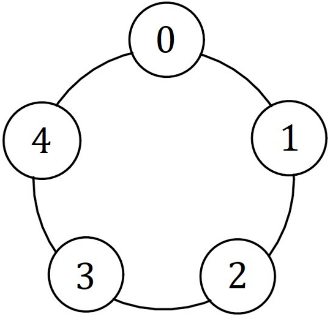
Physical and Virtual Neighbours



Physical Network

$P_3 = f0; 4; 2g$

$V_3 = f4; 2g$



Virtual Network

Table of Contents

Routing Table

Definition 2

The **routing table** at node $u\ 2\ V$ is a collection of 4-tuples (from, to, next hop, prev. hop) from $V \quad V \quad V \quad V$ of routing paths that go through node u.

Routing Table

Definition 2

The **routing table** at node u 2 V is a collection of 4-tuples (from, to, next hop, prev. hop) from $V \quad V \quad V \quad V$ of routing paths that go through node u.

- Each such 4-tuple is called an "entry" in the routing table of u

Routing Table

Definition 2

The **routing table** at node u 2 V is a collection of 4-tuples (from, to, next hop, prev. hop) from $V \quad V \quad V \quad V$ of routing paths that go through node u.

- Each such 4-tuple is called an "entry" in the routing table of u
- The prev. hop and next hop nodes must be physical neighbours of u

Routing Table

Definition 2

The **routing table** at node u 2 V is a collection of 4-tuples (from, to, next hop, prev. hop) from $V \quad V \quad V \quad V$ of routing paths that go through node u.

- Each such 4-tuple is called an "entry" in the routing table of u
- The prev. hop and next hop nodes must be physical neighbours of u
- There is one (and ONLY one) entry in the routing table of u for any path that starts, ends, or passes through u

Routing Table

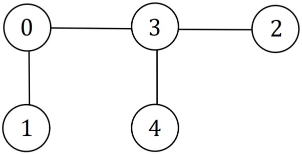
Definition 2

The **routing table** at node u 2 V is a collection of 4-tuples (from, to, next hop, prev. hop) from $V \quad V \quad V \quad V$ of routing paths that go through node u.

- Each such 4-tuple is called an "entry" in the routing table of u
- The prev. hop and next hop nodes must be physical neighbours of u
- There is one (and ONLY one) entry in the routing table of u for any path that starts, ends, or passes through u
- End_u denotes the set of all nodes in the "to" or "from" coordinate in an entry on the routing table of u other than u itself

  i.e. all the nodes we know how to travel to from u

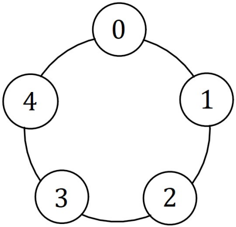
For Example...



Physical Network

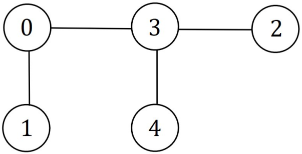
Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

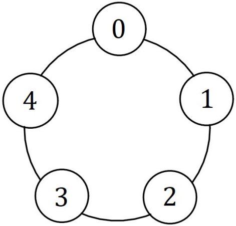


Virtual Network

For Example...



Physical Network



Virtual Network

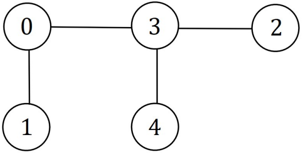
Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

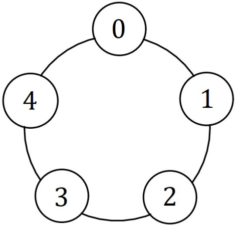
Routing Table at 2

from	to	next hop	prev. hop
2	3	3	-
2	1	3	-

For Example...



Physical Network



Virtual Network

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

Routing Table at 2

from	to	next hop	prev. hop
2	3	3	-
2	1	3	-

Once the physical network is fixed, the routing table is fixed

Table of Contents

The Routing Algorithm

The information each node has access to:

- Its own routing table
- The node to id mapping

The Routing Algorithm

The information each node has access to:

- Its own routing table
- The node to id mapping

We can always route along the ring to our target.

The Routing Algorithm

The information each node has access to:

- Its own routing table
- The node to id mapping

We can always route along the ring to our target.

Suppose we want to send a message from node 2 to node 0.

What should $h2; 0i$ be?

The Routing Algorithm

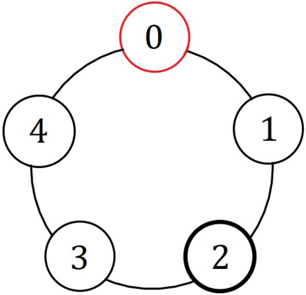
The information each node has access to:

- Its own routing table
- The node to id mapping

We can always route along the ring to our target.

Suppose we want to send a message from node 2 to node 0.

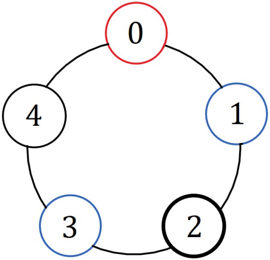
What should *h*2; 0*i* be?



Virtual Network

First Attempt

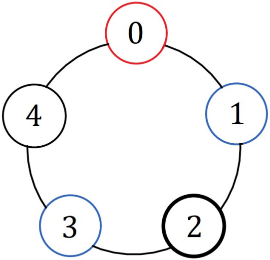
$V_2 = f3; 1g$



First Attempt

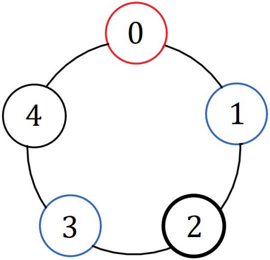
$V_2 = f3;1g$

$d(1;0) < d(3;0) \;=) \;\; T^0 = 1$

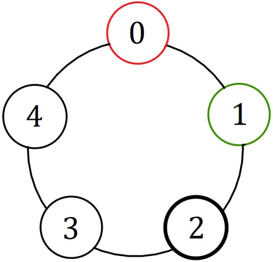


First Attempt

$V_2 = f3; 1g$

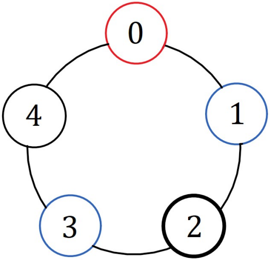


$d(1; 0) < d(3; 0) \;\; =) \;\; T^0 = 1$

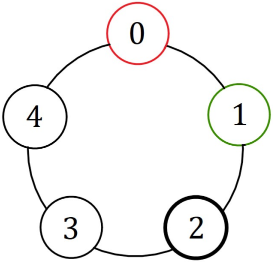


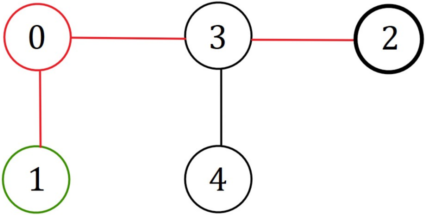
First Attempt

$V_2 = \{3; 1\}$

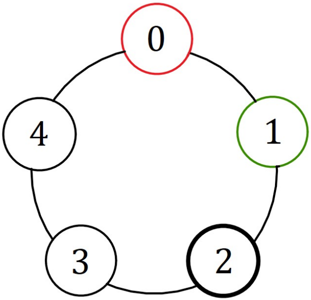


$d(1;0) < d(3;0) \implies T' = 1$

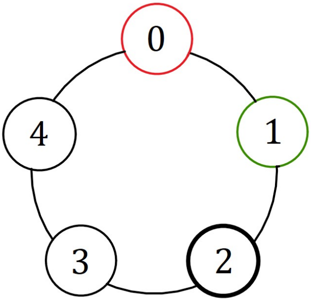


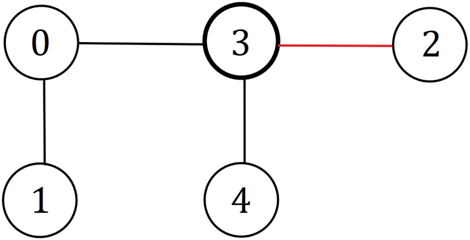


Second Attempt?



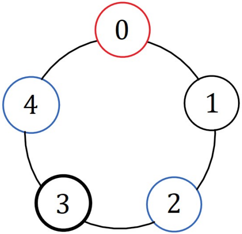
Second Attempt?





Comparing at Each Node

$V_3 = f4; 1g$



Comparing at Each Node

$V_3 = \{4, 1\}$ $\quad$ $d(4, 0) < d(2, 0) \implies T' = 4$

Comparing at Each Node

$V_3 = \{4; 1\}$

$d(4; 0) < d(2; 0) \implies T' = 4$

Comparing at Each Node

$V_3 = \{4; 1\}$

$d(4; 0) < d(2; 0) \implies T' = 4$

Comparing at Each Node

$V_3 = \{4; 1\}$ $\quad d(4;0) < d(2;0) \implies T' = 4$

Third (And Hopefully Last) Attempt

Using the Routing Table

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

Using the Routing Table

$End_3 = \{0; 1; 2; 4\}$

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

The Actual Routing Path

Routing Table Entry for 0

from	to	next hop	prev. hop
4	0	0	4

The Actual Routing Path

Routing Table Entry for 0

from	to	next hop	prev. hop
4	0	0	4

Greedy Transition Vs. Non-Greedy Transition

Greedy Transition Vs. Non-Greedy Transition

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

$V_3 = \{4; 2\}$, $T^0 = 0$

Greedy Transition Vs. Non-Greedy Transition

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

$V_3 = f\,4; 2g,\ T^0 = 0$

Routing Table at 2

from	to	next hop	prev. hop
2	3	3	-
2	1	3	-

$V_2 = f\,1; 3g,\ T^0 = 1$

Greedy Transition Vs. Non-Greedy Transition

Routing Table at 3

from	to	next hop	prev. hop
2	3	-	2
4	3	-	4
0	3	-	0
4	0	0	4
1	2	2	0

$V_3 = \{4, 2\}$, $T' = 0$

Routing Table at 2

from	to	next hop	prev. hop
2	3	3	-
2	1	3	-

$V_2 = \{1, 3\}$, $T' = 1$

$T' \notin V_3$

Greedy Transition

$T' \in V_2$

Non-Greedy Transition

Summarizing The Routing Algorithm

Given a source-target pair s and t, we can summarize the routing algorithm as follows:

Summarizing The Routing Algorithm

Given a source-target pair s and t, we can summarize the routing algorithm as follows:

- At the source node s, we pick an intermediate target $T^0 = x_1$ from End_s and forward the message to the next hop node u towards x_1

Summarizing The Routing Algorithm

Given a source-target pair s and t, we can summarize the routing algorithm as follows:

- At the source node s, we pick an intermediate target $T' = x_1$ from End_s and forward the message to the next hop node u towards x_1
- At u, we consider all the elements of End_u. If we find a better node $x_2 \in End_u$, then we change the intermediate target and set $T' = x_2$ and forward to the next hop node towards x_2

Summarizing The Routing Algorithm

Given a source-target pair s and t, we can summarize the routing algorithm as follows:

- At the source node s, we pick an intermediate target $T^0 = x_1$ from End_s and forward the message to the next hop node u towards x_1
- At u, we consider all the elements of End_u. If we find a better node $x_2 \in End_u$, then we change the intermediate target and set $T^0 = x_2$ and forward to the next hop node towards x_2
- If no such x_2 exists in End_u, then we forward the message along to the next hop node towards $T^0 = x_1$

Summarizing The Routing Algorithm

Given a source-target pair s and t, we can summarize the routing algorithm as follows:

- At the source node s, we pick an intermediate target $T' = x_1$ from End_s and forward the message to the next hop node u towards x_1
- At u, we consider all the elements of End_u. If we find a better node $x_2 \in End_u$, then we change the intermediate target and set $T' = x_2$ and forward to the next hop node towards x_2
- If no such x_2 exists in End_u, then we forward the message along to the next hop node towards $T' = x_1$
- Once we reach an intermediate target x_i, we set the intermediate target from End_{x_i}

What exactly is a better node?

x is a better node if $d(x; T) < d(T'; T)$, where T is the target, T' is the intermediate target

Note: We want to choose the "best" possible x, i.e. the one that gets us closest to the target

Some Edge Cases

Some Edge Cases

Remarks About the Intermediate Target(s)

- Each new intermediate target T' is always closer to the target T than the previous one.
- At any node u, the current intermediate target is closer to the target than u or equidistant
- Once we set $T' = T$, there are no more new intermediate targets

Properties of This Routing Algorithm

Proposition 1

VRR schemes using greedy routing exhibit no loops

Proof: If we have a loop, we must visit the same node twice, say v.

Properties of This Routing Algorithm

Proposition 1

VRR schemes using greedy routing exhibit no loops

Proof: If we have a loop, we must visit the same node twice, say

At time 1

Properties of This Routing Algorithm

Proposition 1

VRR schemes using greedy routing exhibit no loops

Proof: If we have a loop, we must visit the same node twice, say v.

At time 1

At time 2

Table of Contents

Bene ts of VRR

- Each node only stores local info; less memory used relative to increase in size of physical network
- Flexible since adding/ removing a node only a ects nodes that had that node in their routing table

Limitations of VRR

Routing path may not be the shortest path

Limitations of VRR

Routing path may not be the shortest path

De nition 3

We de ne stretch as the ratio between the actual routing path and the shortest possible path on the physical network

Limitations of VRR

Routing path may not be the shortest path

De nition 3

We de ne stretch as the ratio between the actual routing path and the shortest possible path on the physical network

Consider the following example

Stretch

Suppose again that we are routing $\langle 2, 0 \rangle$. Our ring:

The Actual Routing Path

from	to	next hop	prev. hop
2	3	3	-
2	1	3	-

$d(1;0) < d(3;0)$

=) $T^0 = 1$

The Actual Routing Path

from	to	next hop	prev. hop
3	5	5	-
3	4	4	-
3	2	2	-
2	1	4	2
4	5	5	4

$d(5;0)$ $d(1;0)$

=) $T^{0} = 1$

The Actual Routing Path

from	to	next hop	prev. hop
4	3	3	-
4	1	2	-
4	5	3	-

$d(5;0) \quad d(1;0)$

$=) \quad T^0 = 1$

Stretch

$L_A = 5$

Stretch

$L_A = 5$

$L_S = 3$

Stretch

$L_A = 5$

$L_S = 3$

Stretch is 5:3

Table of Contents

Current Research: Careful Study of Stretch

The questions:

How can we study stretch?

Current Research: Careful Study of Stretch

The questions:

- How can we study stretch?
- How does stretch change as $N \to \infty$

Current Research: Careful Study of Stretch

The questions:

- How can we study stretch?
- How does stretch change as N ! 1
- What other factors might a ect stretch?

Current Research: Careful Study of Stretch

The questions:

- How can we study stretch?
- How does stretch change as $N \to \infty$
- What other factors might affect stretch?

The issues:

- Setting up a good model for the physical network

Current Research: Careful Study of Stretch

The questions:

- How can we study stretch?
- How does stretch change as $N \to \infty$
- What other factors might affect stretch?

The issues:

- Setting up a good model for the physical network
- Computing the expected path length

The virtual network for large N

Table of Contents

Physical Network as a Ring

When the physical network is also a ring, we call it a physical ring.

Physical Network as a Ring

When the physical network is also a ring, we call it a physical ring.

There are only two physical neighbours

Physical Network as a Ring

- When the physical network is also a ring, we call it a physical ring.
- There are only two physical neighbours
- We can consider the physical ring a permutation of the virtual ring

Stretch on a Ring

Proposition 2

If the physical network is a ring of $N > 3$ nodes, then the maximum stretch for any routing path is $\frac{N-2}{2}$

Proof of Prop'n 2

Proof: Consider source-target pair S and T. Suppose that, on the physical ring, $d(S;T) = k \quad 2$

Figure: The Physical Ring

Proof of Prop'n 2

Proof: Consider source-target pair S and T. Suppose that, on the physical ring, $d(S; T) = k \quad 2$

Figure: The Physical Ring

By prop'n 1, the actual routing path either travels the minor arc or the major arc.

Proof Cont'd

If we travel the minor arc then

$$\frac{L_A}{L_S} = \frac{k}{k} = 1$$

Figure: The Physical Ring

Proof Cont'd

If we travel the minor arc then

$$\frac{L_A}{L_S} = \frac{k}{k} = 1$$

If we travel the major arc then

$$\frac{L_A}{L_S} = \frac{N \quad k}{k}$$

Figure: The Physical Ring

Proof Cont'd

If we travel the minor arc then

$$\frac{L_A}{L_S} = \frac{k}{k} = 1$$

If we travel the major arc then

$$\frac{L_A}{L_S} = \frac{N \quad k}{k}$$

This ratio is maximized when

$k = 2$

Figure: The Physical Ring

Table of Contents

Limitations of the Physical Ring

Not practical since relatively poor connectivity

Limitations of the Physical Ring

Not practical since relatively poor connectivity

Even the shortest paths are long

Future Directions

Explore other physical network models

Future Directions

- Explore other physical network models
- Trying to formulate routing algorithm into Markov Chain process or as (sub)martingales and apply related theorems

The End

Thanks for listening!
Any questions?

For Further Reading

To better understand the mechanics of VRR:

M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron. Virtual ring routing: Network routing inspired by DHTs. In *ACM annual conference of the Special Interest Group on Data Communication (SIGCOMM)*, pages 351–362, 2006.

To read about other proofs related to VRR:

Malkhi, D., Sen, S., Talwar, K., Werneck, R. F., and Wieder, U. (2009). Virtual ring routing trends. In *Proceedings of the 23rd international conference on distributed computing*, Berlin, Heidelberg.