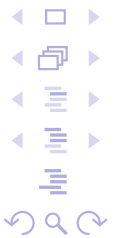
Compact Routing

Michael Dom

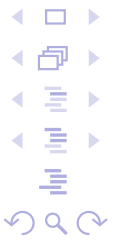
Friedrich-Schiller-Universität Jena

GI-Dagstuhl-Forschungsseminar Nr. 05473,
Algorithmen für Sensor- und Ad-hoc-Netzwerke,
23.-25. November 2005,
IBFI Schloss Dagstuhl

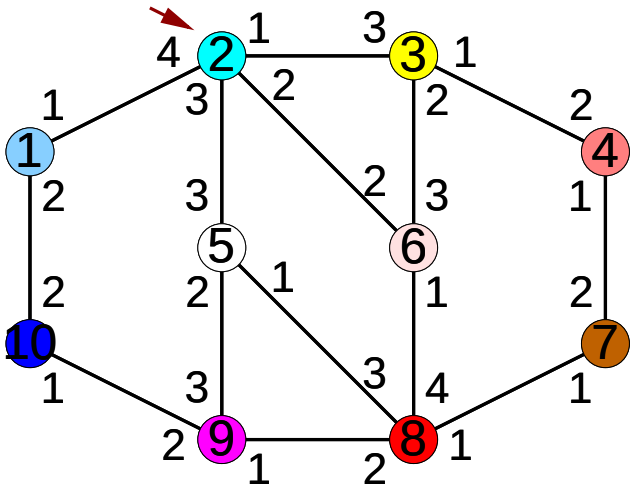
Outline



Outline

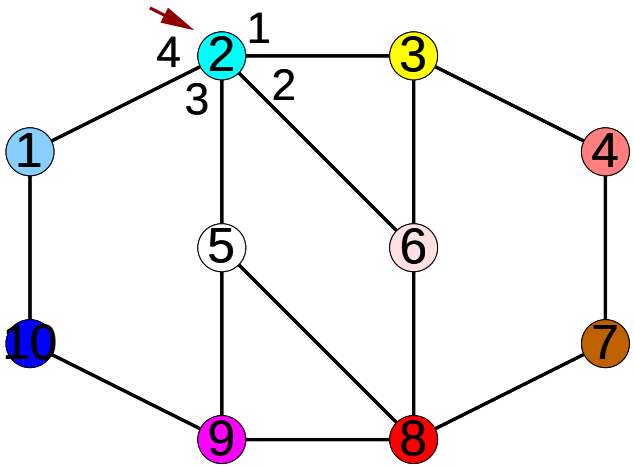


Routing: First Examples



7
DATA

Routing: First Examples

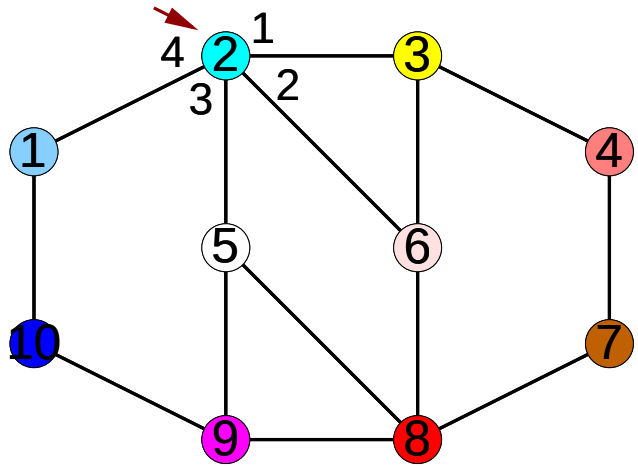


7
DATA

Local memory of node 2**:**	
Own address:	2
Port for node 1:	4
Port for node 3:	1
Port for node 4:	1
Port for node 5:	3
Port for node 6:	2
Port for node 7:	1
Port for node 8:	2
Port for node 9:	3
Port for node 10:	4

- Local memory:
  $n \cdot \log(\deg(v))$ bits

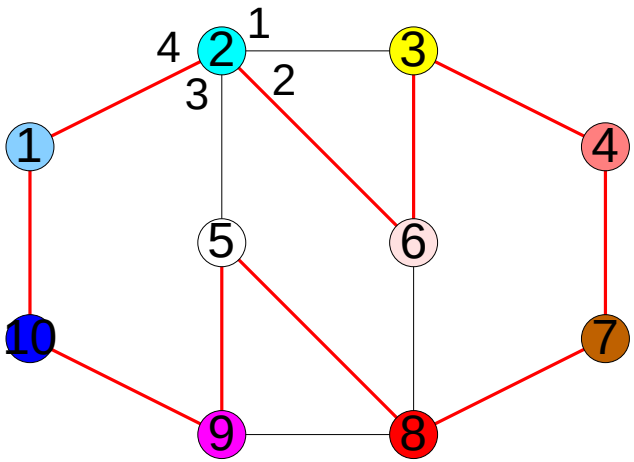
Routing: First Examples



Local memory of node 2:	
Own address:	2
Port 1:	3–4
Port 2:	6–8
Port 3:	5
Port 4:	9–1

- Local memory: $\deg(v) \cdot 2 \log n$ bits

Routing: First Examples



Local memory of node 2**:**	
Own address:	2
Port 2 for every destination.	

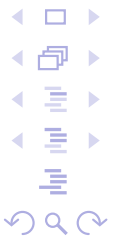
- Local memory:
  $\log n + \log(\deg(v))$ bits

7
DATA

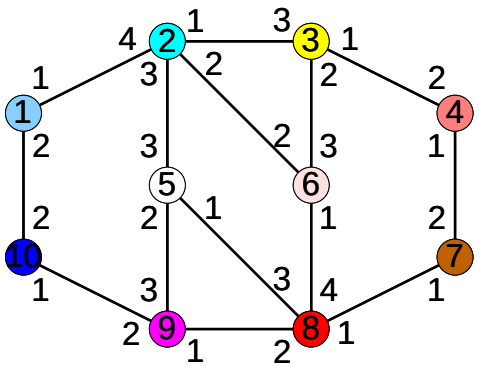
Routing Strategies

Routing Strategy:

- Global preprocessing-algorithm that initializes the data structures.
- Distributed algorithm (“routing scheme”) that runs on every node and performs the routing.



Routing: Models



Addresses:

- Part of the input: Labeled Routing
- Assigned to the nodes by the algorithm: Name Independent Routing

Port Numbers:
Always integers from 1 to $\deg(v)$.

- Part of the input: Fixed Port Model
- Assigned to the edges by the algorithm: Designer Port Model

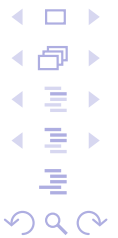
Headers:

- Non-writable/Writable

Performance Measurements

- Local memory size
- Total memory size
- Address/header size
- Stretch (ratio length of routing path / length of shortest path)
- Routing time per node
- Time needed to initialize network

Trade-off: Local memory $\leftrightarrow$ Stretch.
Compact Routing Schemes use $o(n)$ bits of local memory.



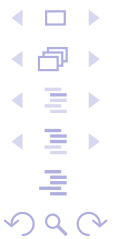
Some Results

Lower bounds:

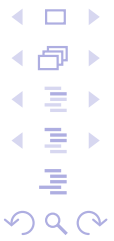
- Stretch $< 3 \Rightarrow \Omega(n)$ bits of local memory [Gavoille and Gengler, SIROCCO 1997].
- Loop-free with addresses from 1 to $n \Rightarrow$ no scheme with $O(\sqrt{n})$ bits of local memory for every stretch, even in trees [Eilam et al., PODC 1998].

Upper bounds:

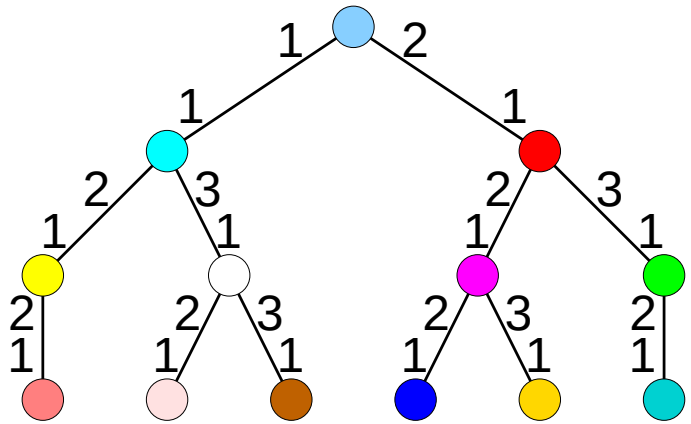
- Shortest path routing in trees with $O((\log n)^2 / \log\log n)$ bits local memory [Thorup and Zwick, SPAA 2001].
- Stretch 3 with $O(\sqrt{n}(\log n)^3 / \log\log n)$ bits local memory [Abraham et al., SPAA 2004].



Outline

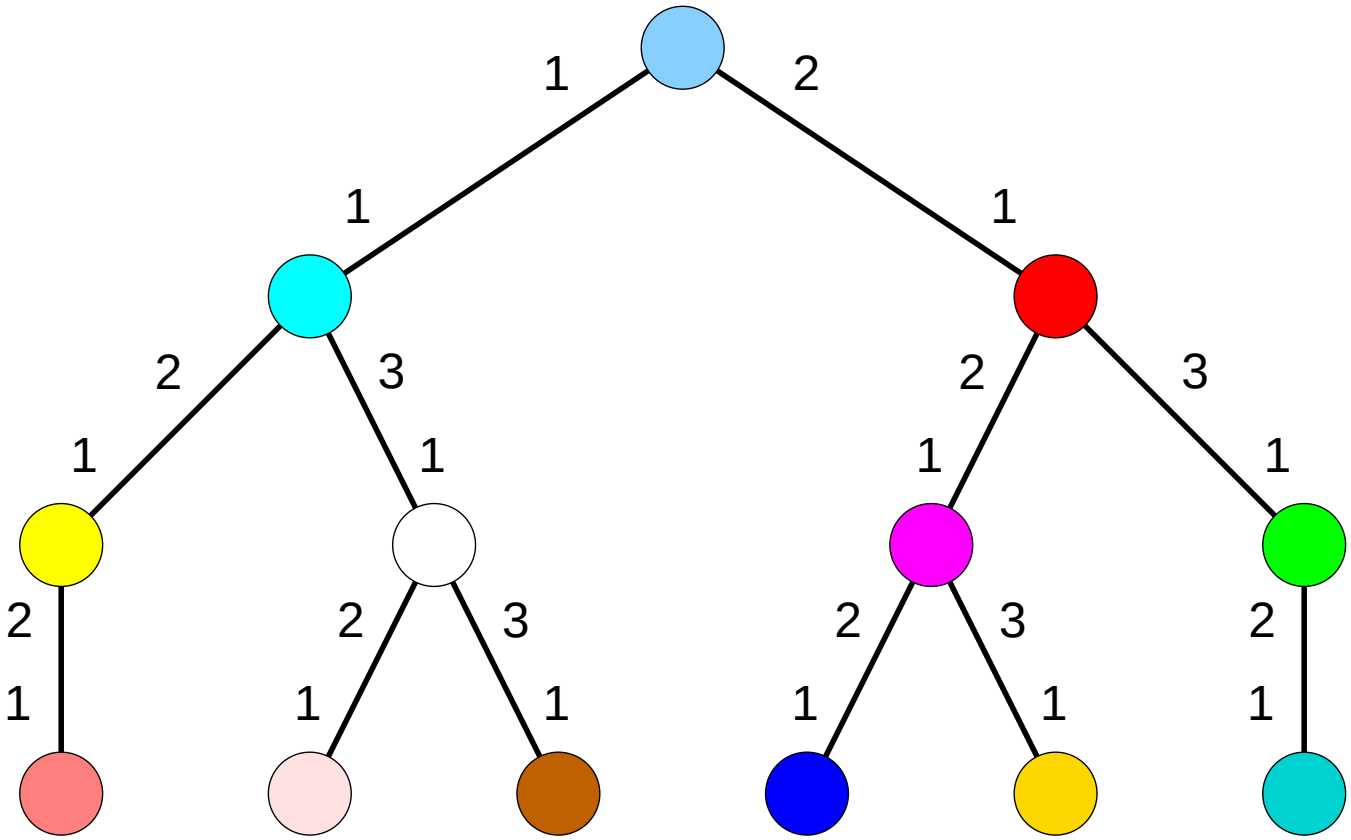


Labeled Routing in Trees

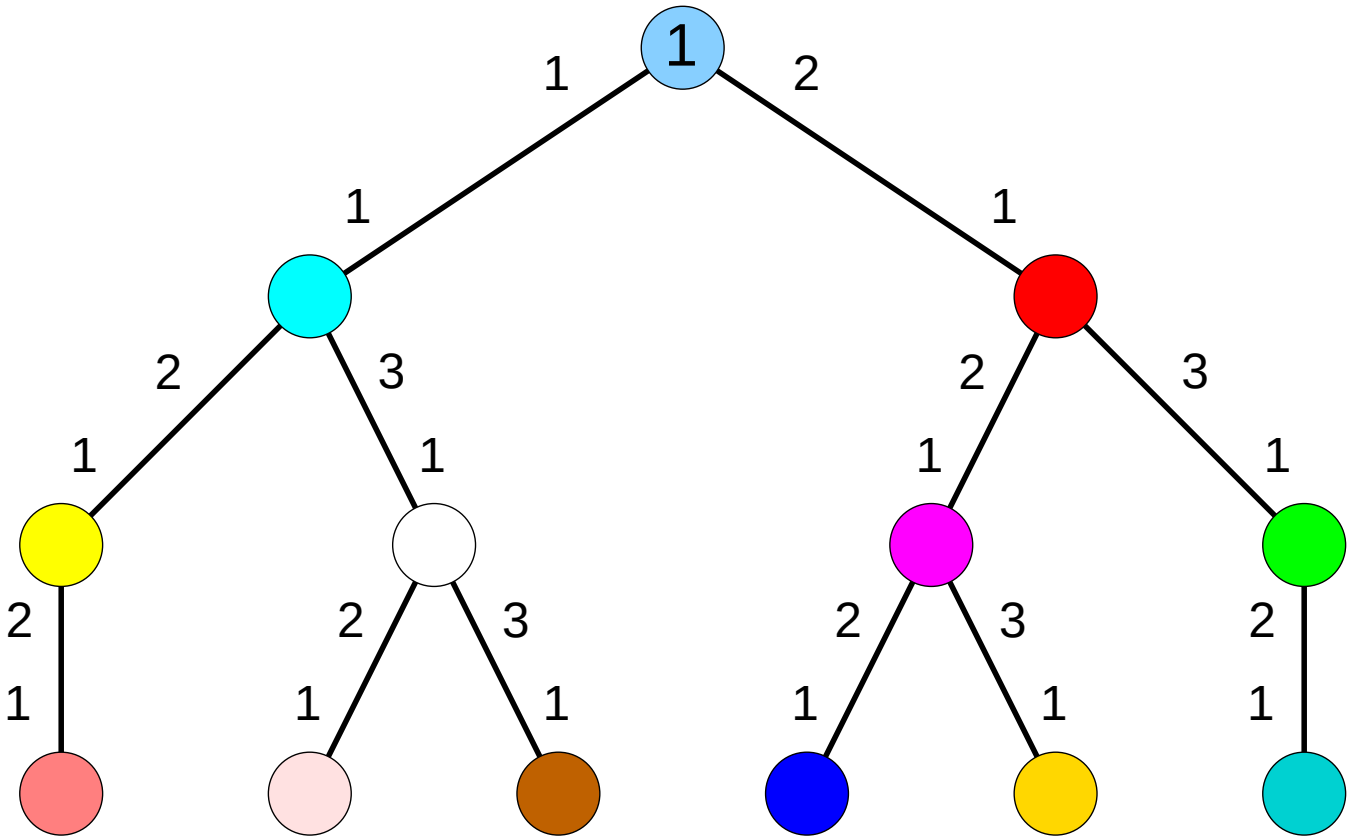


- Interval Routing Scheme by Santoro and Khatib
  [SIAM Journal on Computing, 1985].
- Needs $O(\deg(v) \cdot \log n)$ bits of local memory.

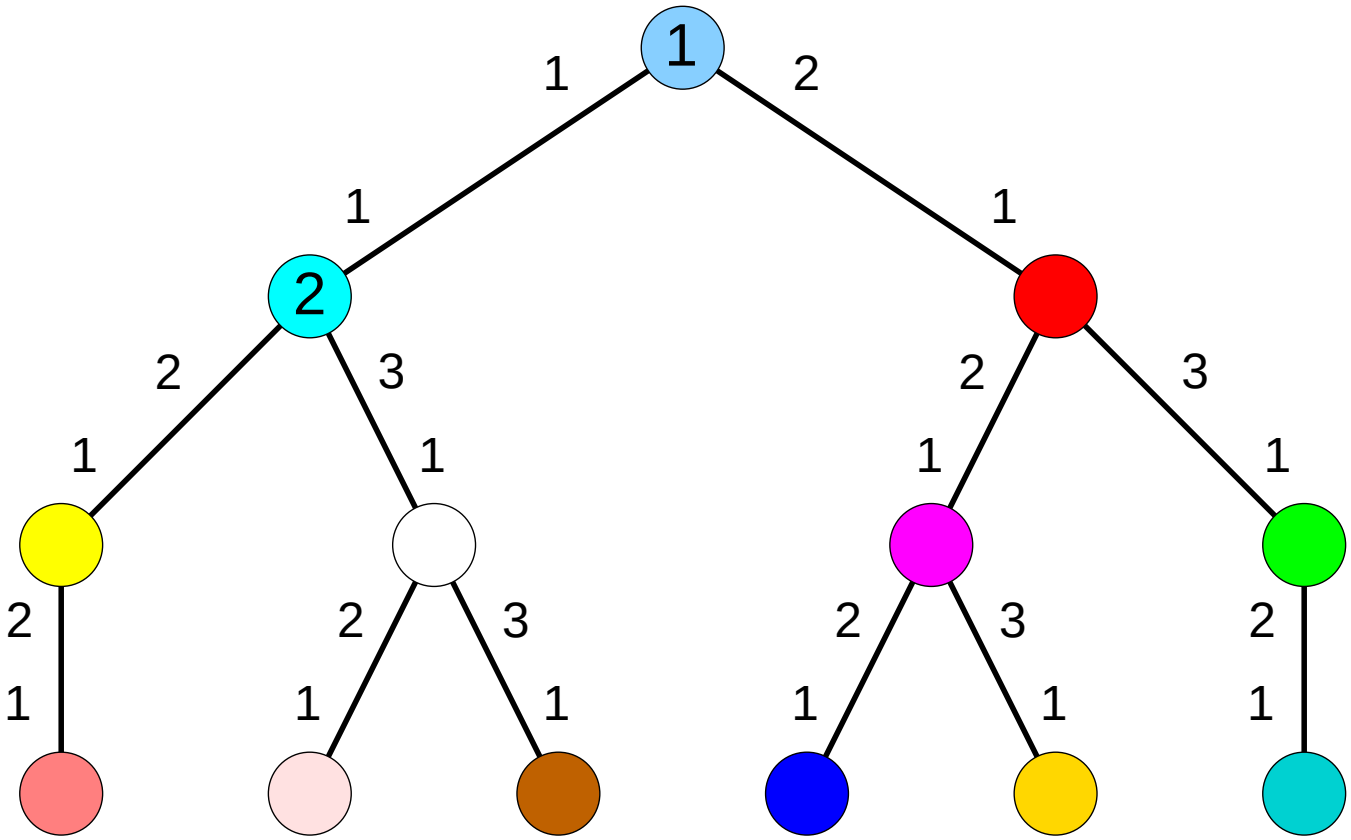
Assigning Addresses to Nodes via DFS



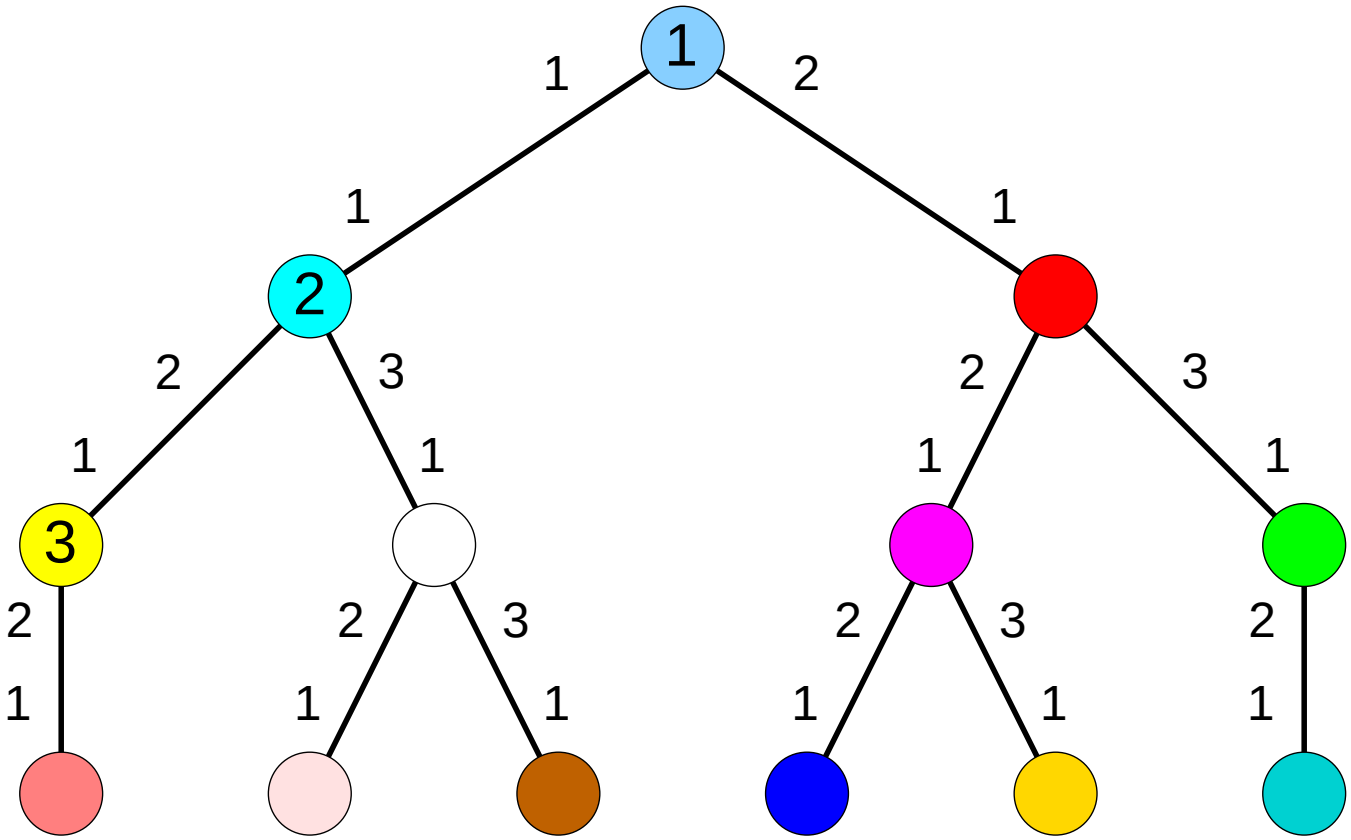
Assigning Addresses to Nodes via DFS



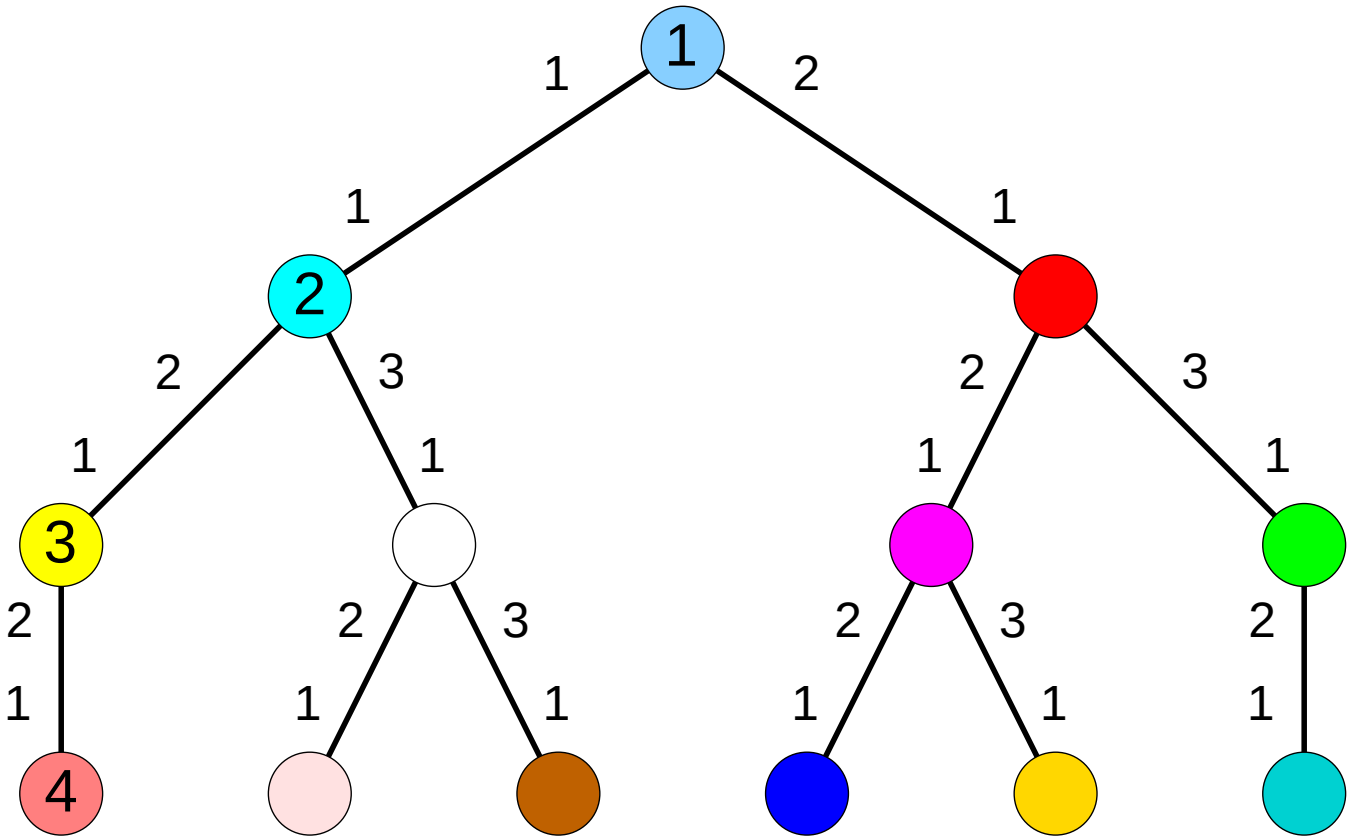
Assigning Addresses to Nodes via DFS



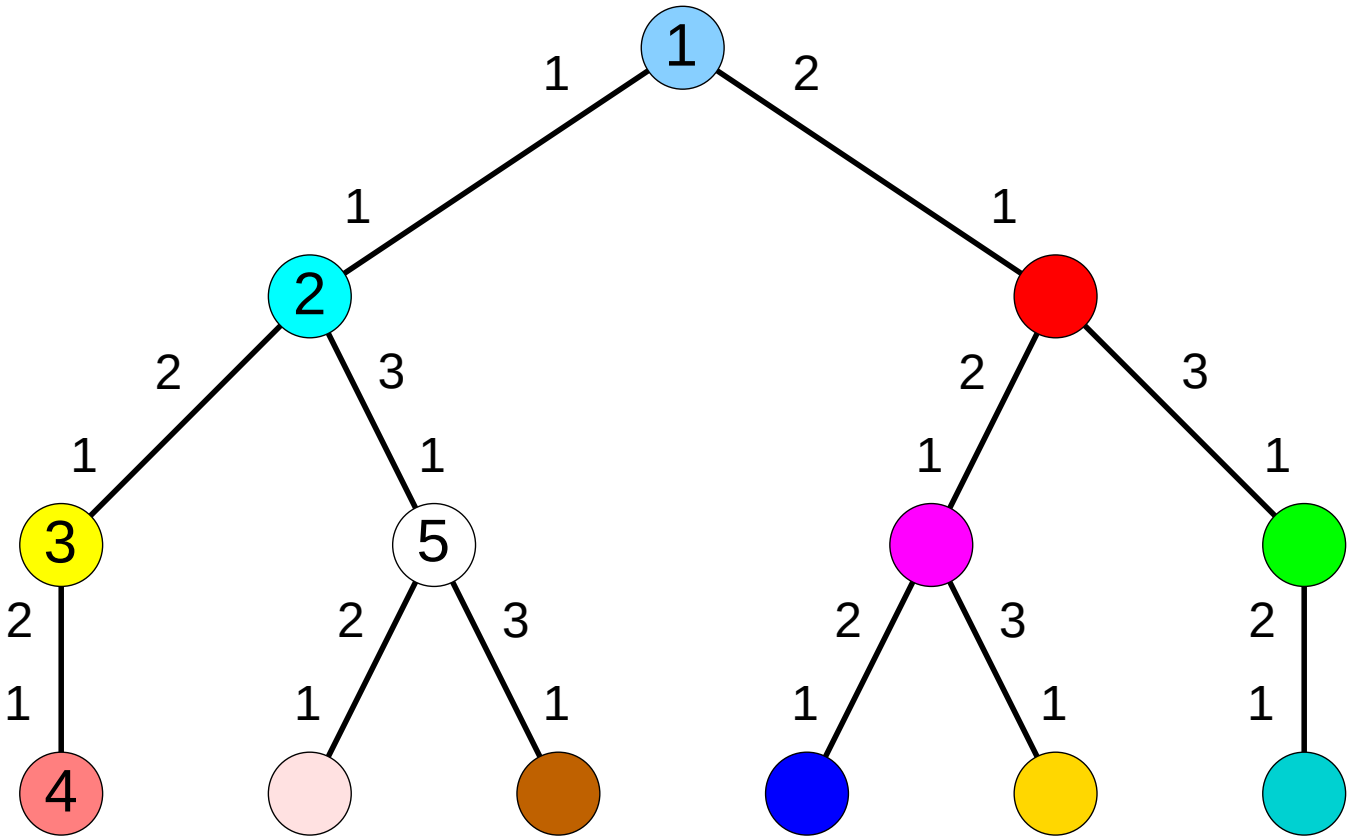
Assigning Addresses to Nodes via DFS



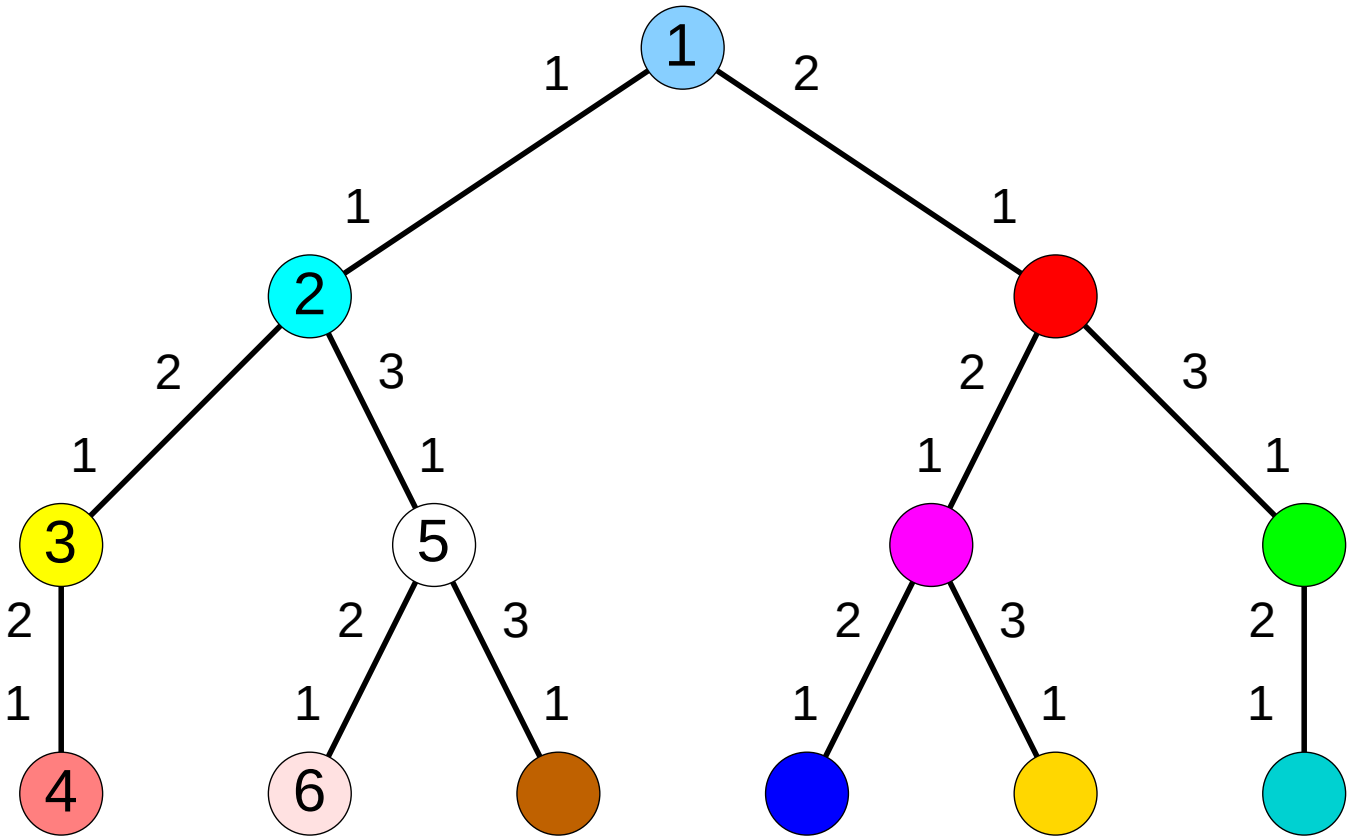
Assigning Addresses to Nodes via DFS



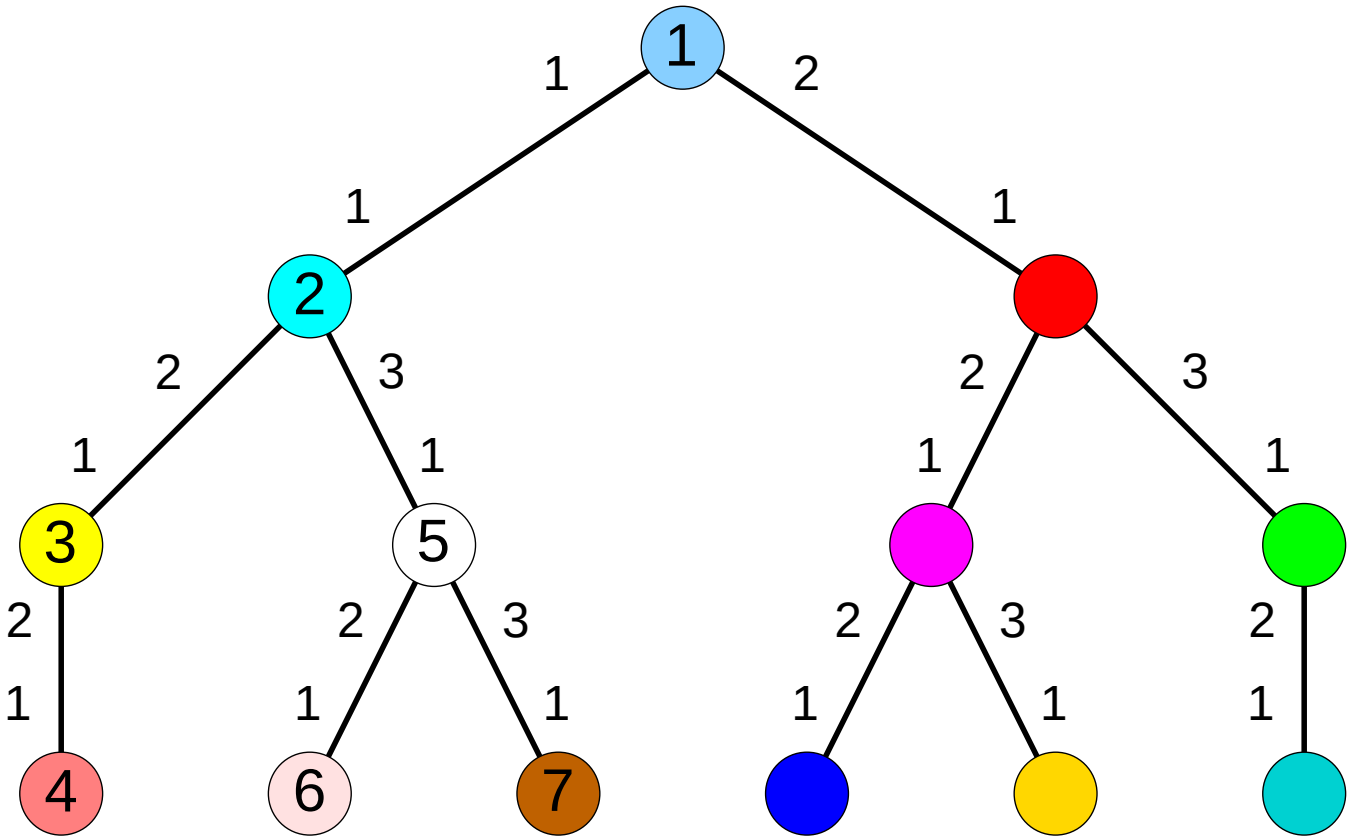
Assigning Addresses to Nodes via DFS



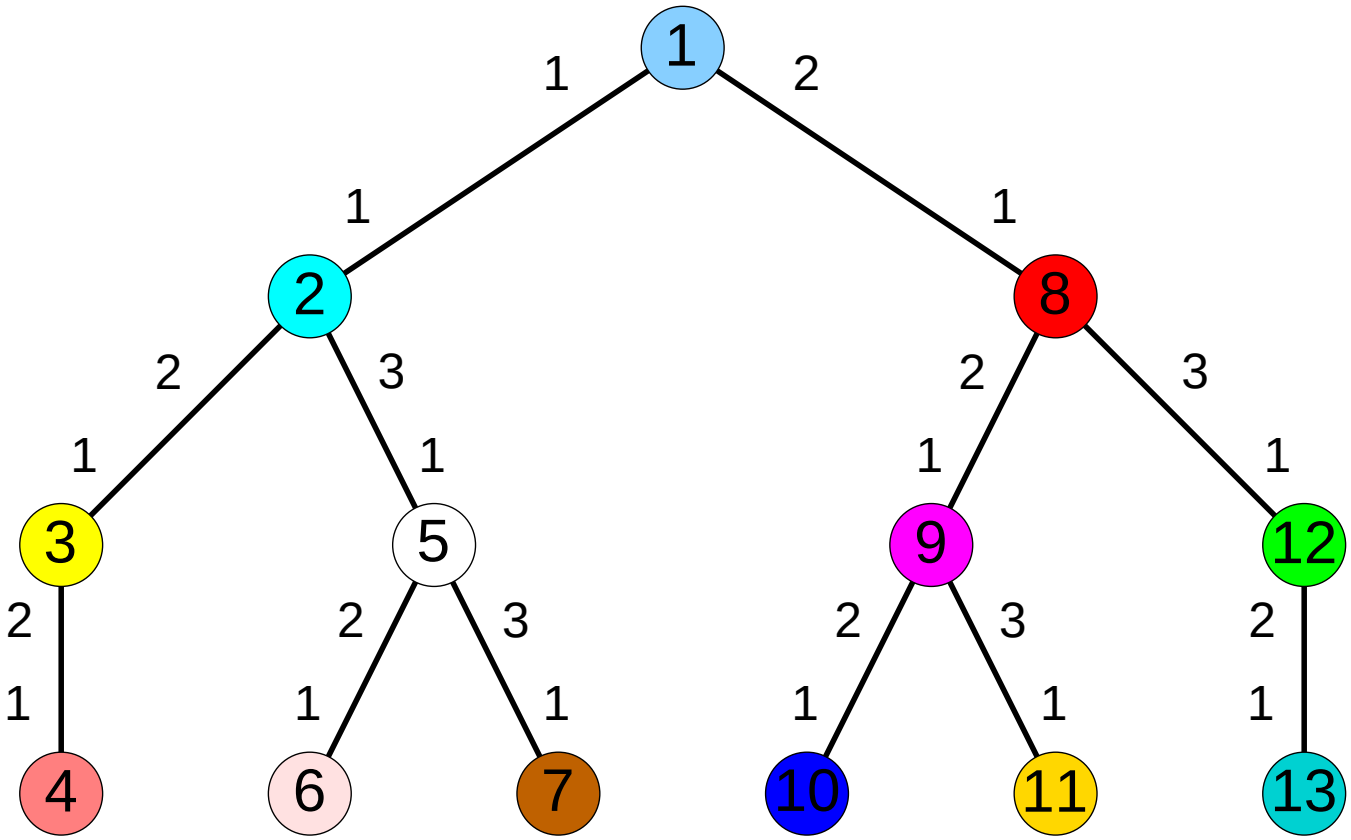
Assigning Addresses to Nodes via DFS



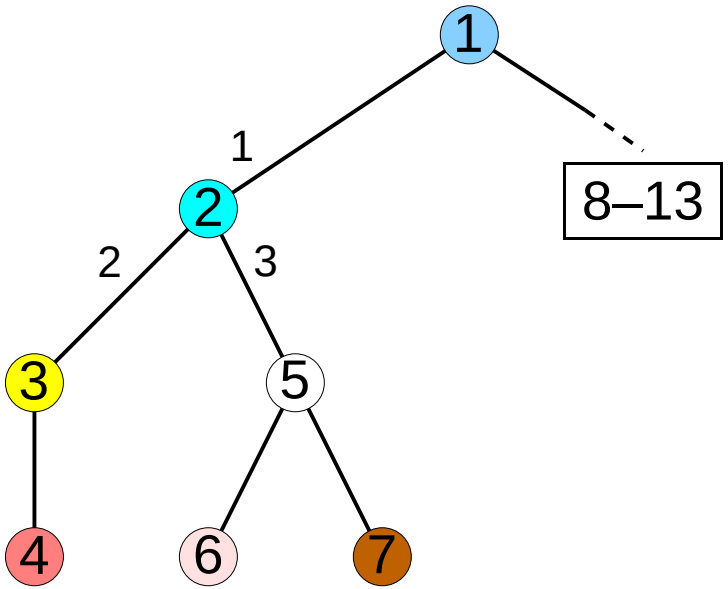
Assigning Addresses to Nodes via DFS



Assigning Addresses to Nodes via DFS

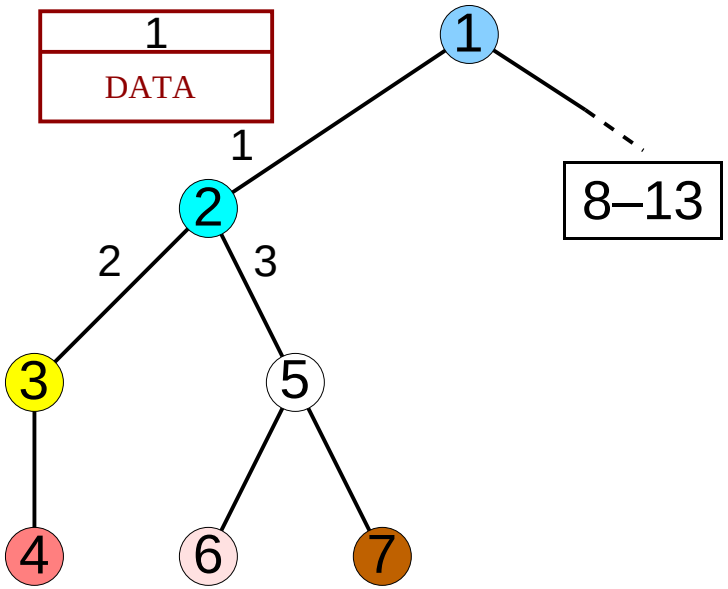


The Local Memory



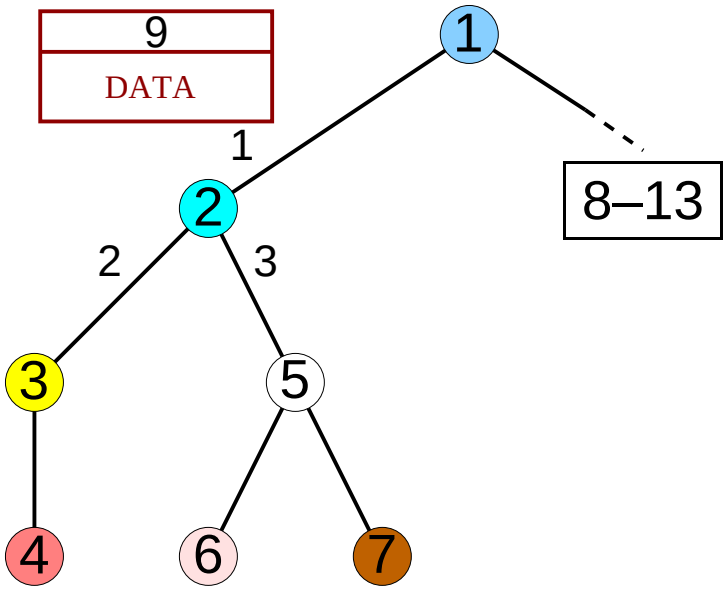
Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

Routing Data Packages



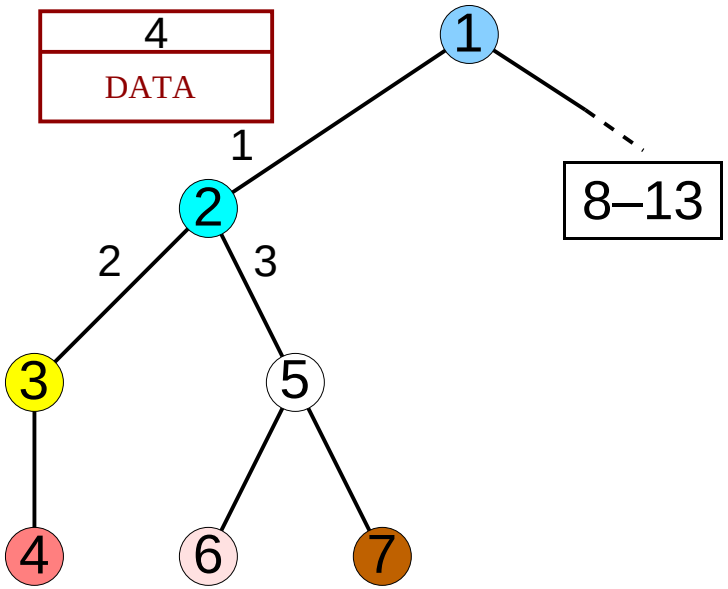
Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

Routing Data Packages



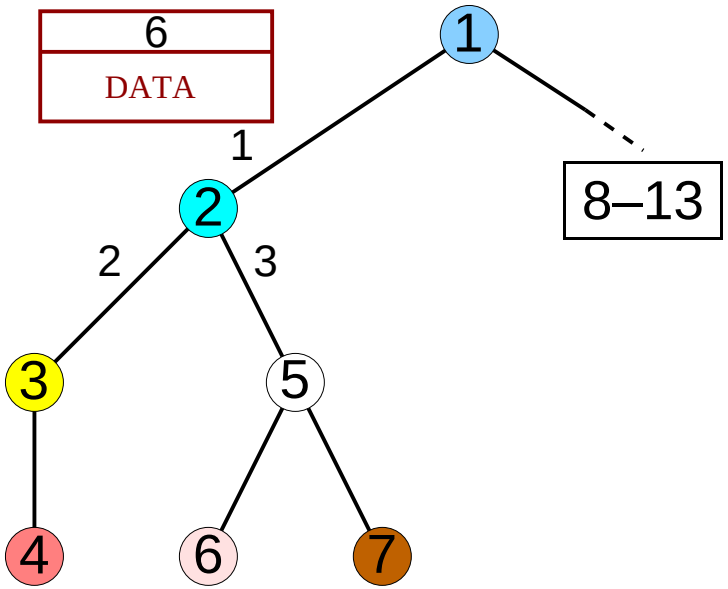
Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

Routing Data Packages



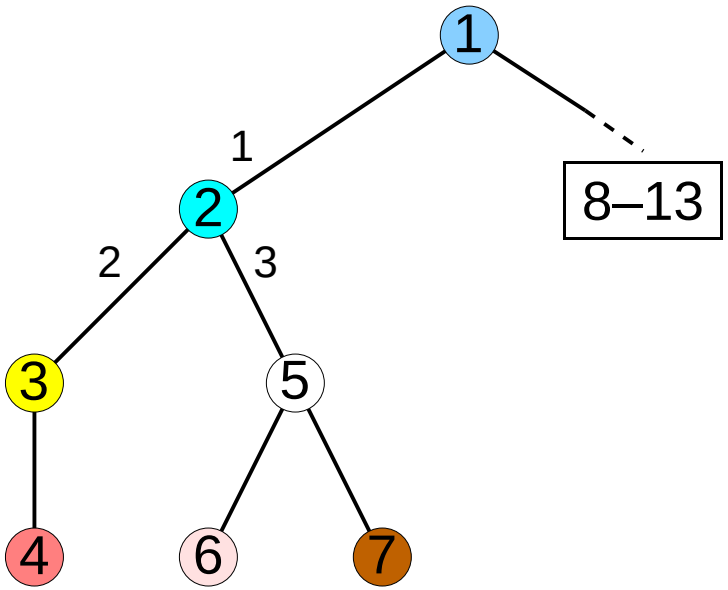
Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

Routing Data Packages



Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

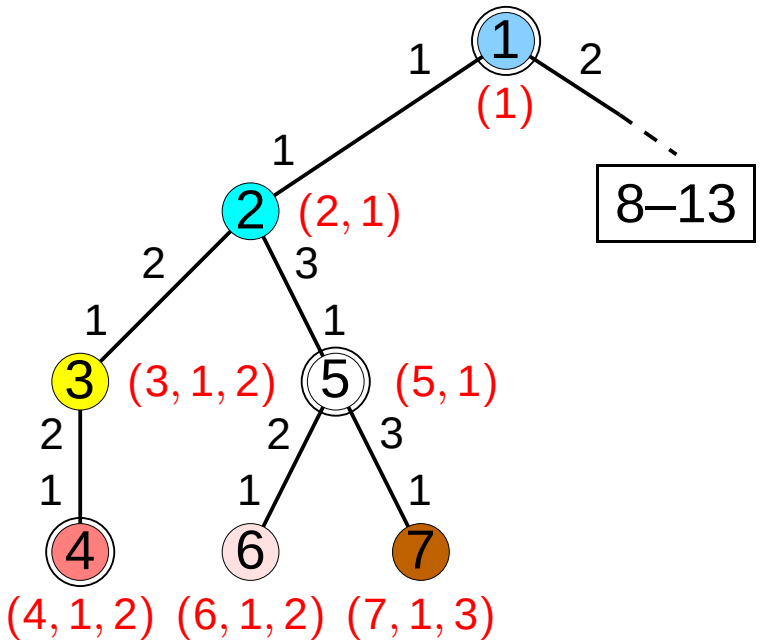
Properties of the Routing Strategy



Local memory of node 2**:**	
Own address:	2
Highest addr. in subtree:	7
Port to parent:	1
Port to subtree:	(4, 2)
Port to subtree:	(7, 3)

- Address size: $\log n$ bits
- Local memory size: $O(\deg(v) \cdot \log n)$ bits
- Can be improved to size of addresses and local memory $O((\log n)^2 / \log\log n)$ bits.

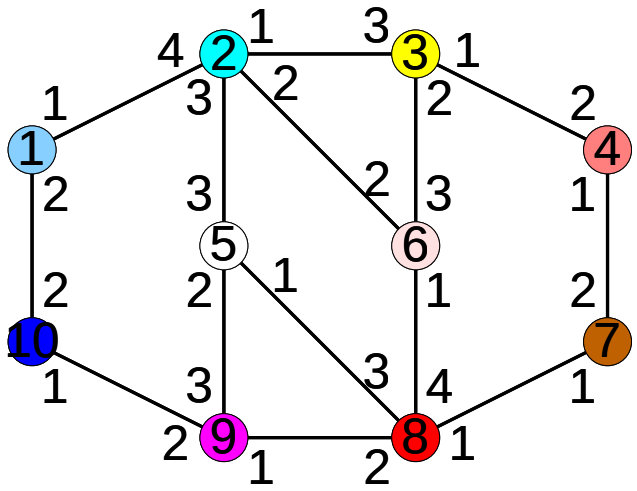
Storing Informations in the Addresses



Local memory of node 2**:**	
Own DFS number:	2
Highest DFS no. in subtree:	7
DFS Number of “heavy” child:	5
#“light” nodes on path from root:	1
Port to parent:	1
Port to heavy child:	3

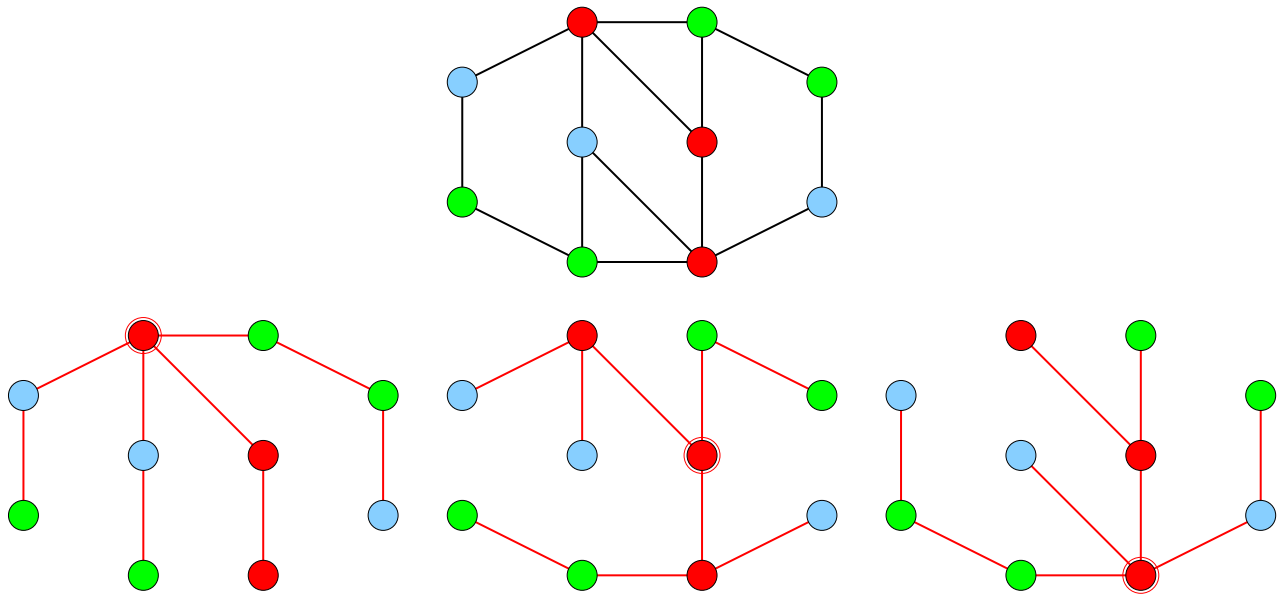
[Thorup and Zwick, SPAA 2001]

A Universal Name Independent Routing Scheme



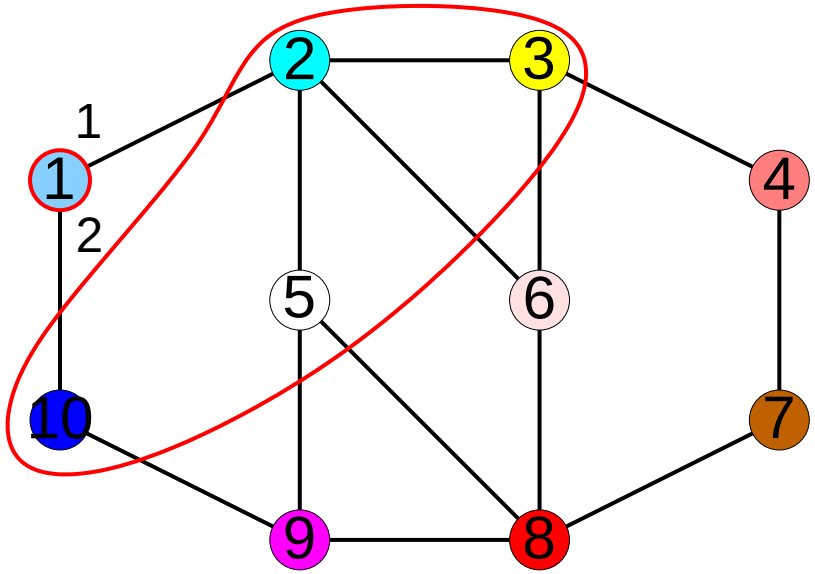
- Scheme of Abraham et al. [SPAA 2004].
- The best known name independent routing scheme with stretch 3.
- Needs $O(\sqrt{n}(\log n)^3 / \log\log n)$ bits of local memory.
- Writable headers of size $O((\log n)^2 / \log\log n)$ bit.

Main idea: Use spanning trees!



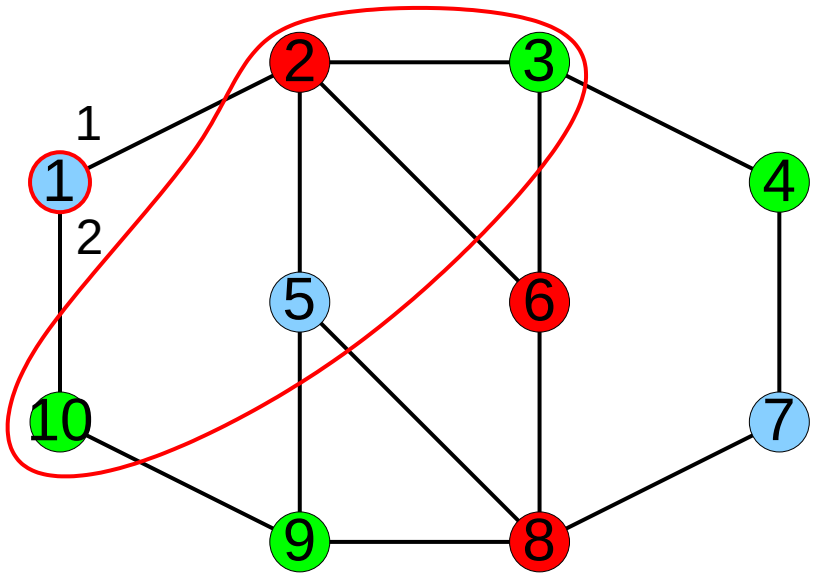
- Several spanning trees.
- For each tree: Routing information of size $O((\log n)^2 / \log \log n)$ bit in local memory.
- Problem: The routing scheme for trees is a labeled routing scheme!

Ingredients



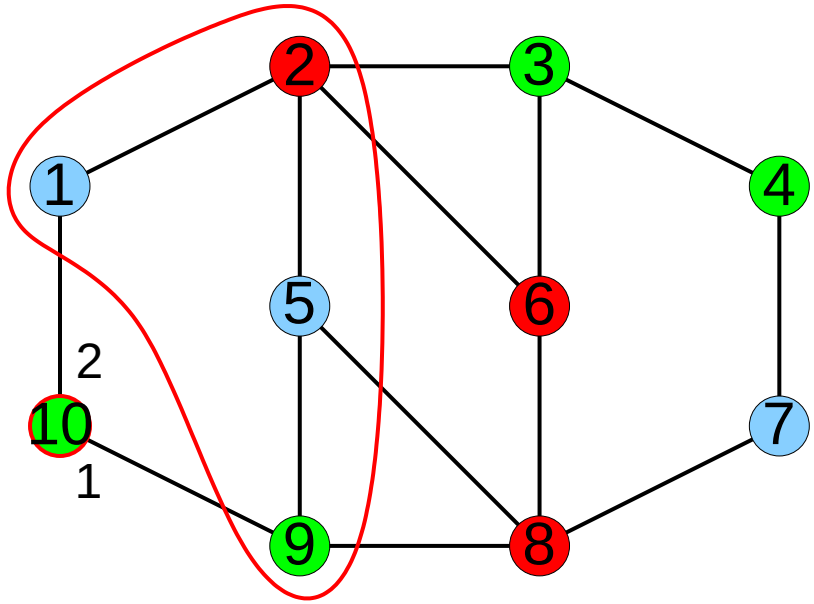
- The *vicinity* $B(v)$ of a node v is the set of the $b\sqrt{n}\log n$ nearest nodes with b being a constant.

Ingredients



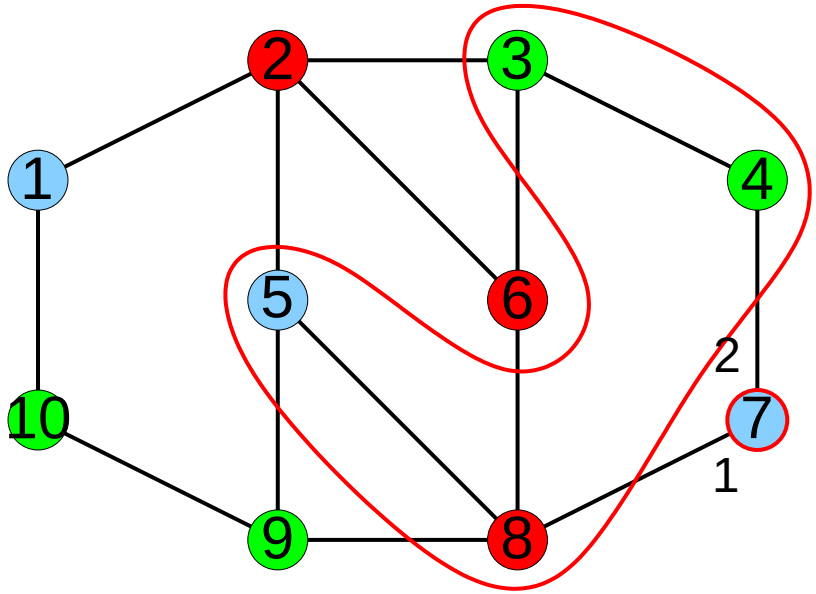
- The *vicinity* $B(v)$ of a node v is the set of the $b\sqrt{n}\log n$ nearest nodes with b being a constant.
- There is a coloring with $\sqrt{n}$ colors such that in each $B(v)$ there is at least one node from each color. There are at most $2\sqrt{n}$ nodes of each color.

Ingredients



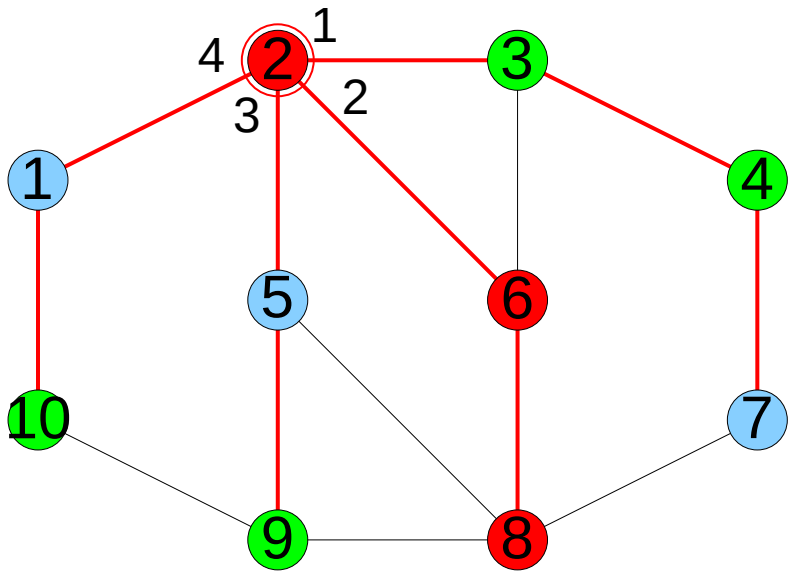
- The *vicinity* $B(v)$ of a node v is the set of the $b\sqrt{n}\log n$ nearest nodes with b being a constant.
- There is a coloring with $\sqrt{n}$ colors such that in each $B(v)$ there is at least one node from each color. There are at most $2\sqrt{n}$ nodes of each color.

Ingredients



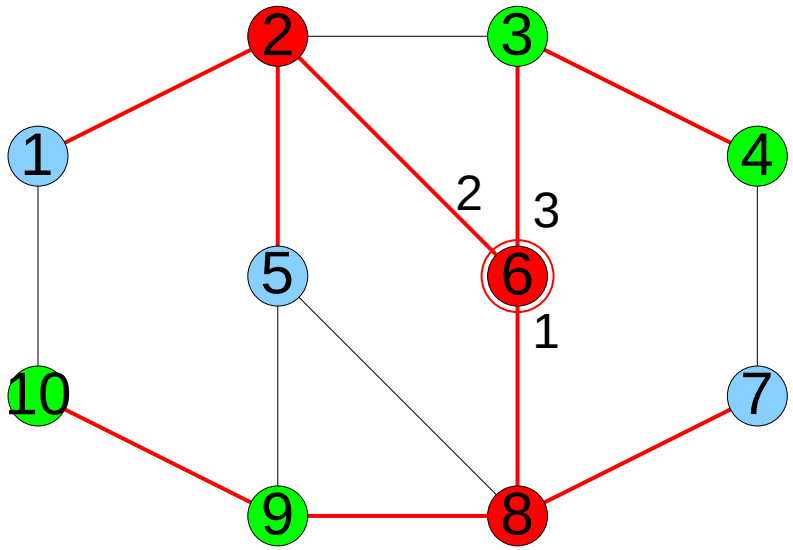
- The *vicinity* $B(v)$ of a node v is the set of the $b\sqrt{n}\log n$ nearest nodes with b being a constant.
- There is a coloring with $\sqrt{n}$ colors such that in each $B(v)$ there is at least one node from each color. There are at most $2\sqrt{n}$ nodes of each color.

Ingredients



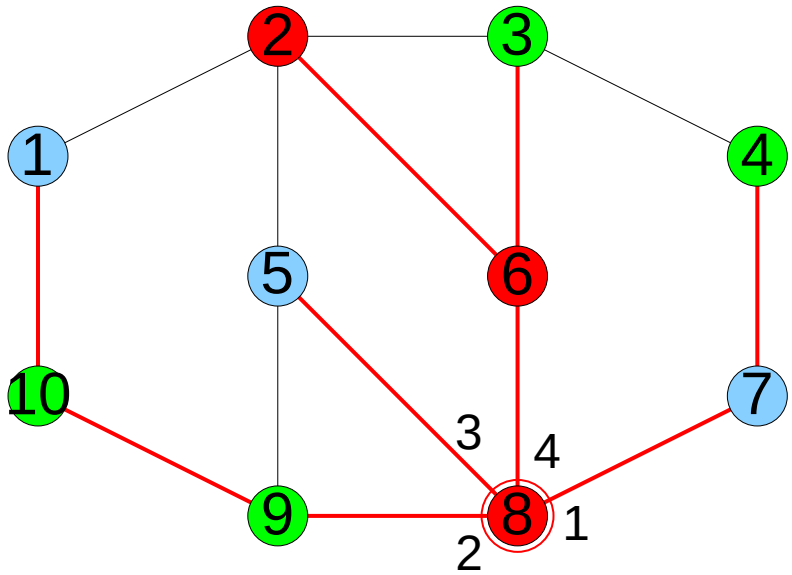
- For each red node i, we denote with T_i the tree consisting of the shortest paths from i to all other nodes.

Ingredients



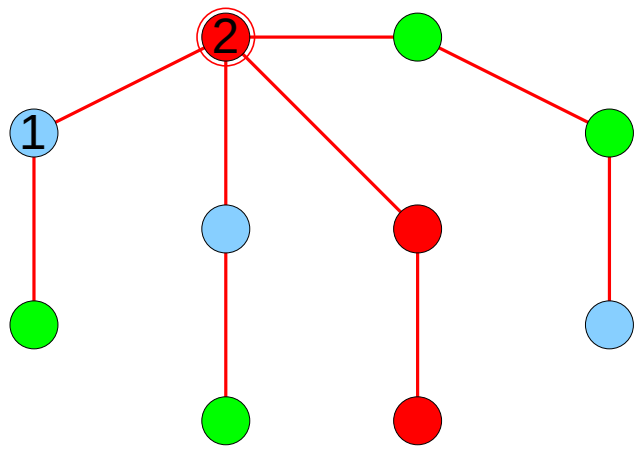
- For each red node i, we denote with T_i the tree consisting of the shortest paths from i to all other nodes.

Ingredients



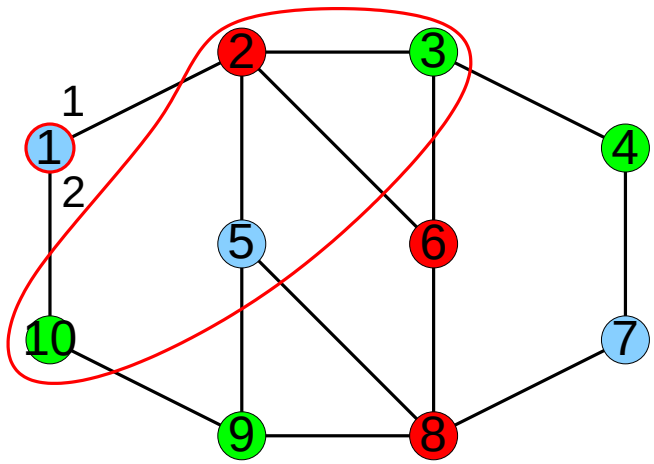
- For each red node i, we denote with T_i the tree consisting of the shortest paths from i to all other nodes.

The Local Memory



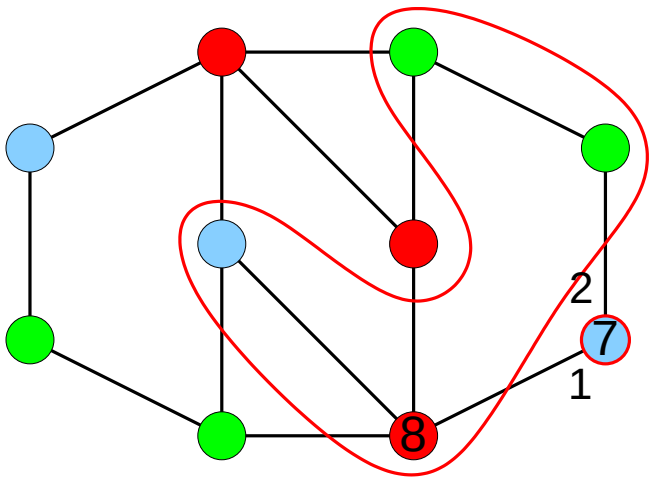
Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

The Local Memory



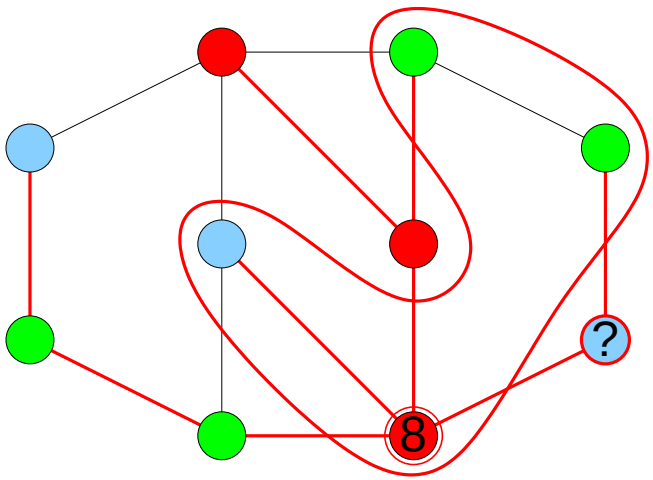
Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

The Local Memory



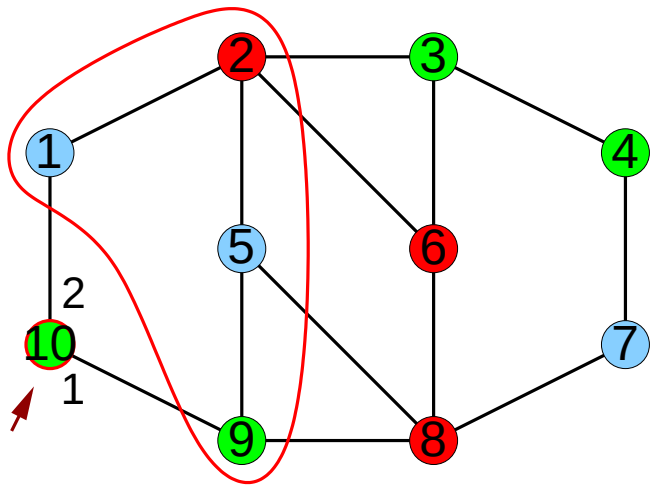
Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

The Local Memory



Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

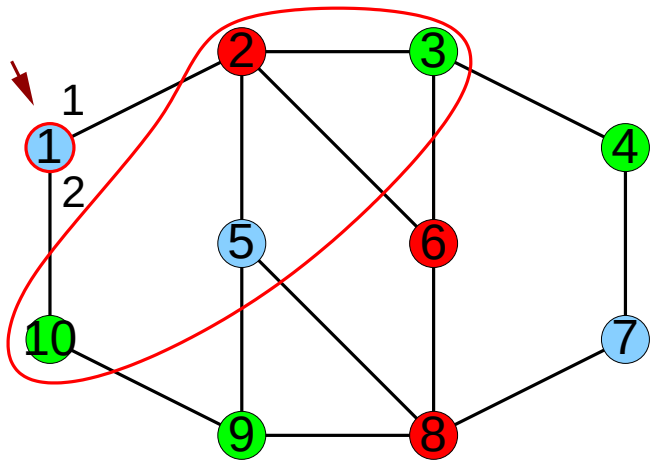
Routing Data Packages



7
DATA

Local memory of node 10:	
Own address:	10
Local memory of node 10 for labeled routing in T_2:	
...	...
...	...
Local memory of node 10 for labeled routing in T_6:	
...	...
...	...
Local memory of node 10 for labeled routing in T_8:	
...	...
...	...
Port for node $1 \in B(10)$:	2
Port for node $2 \in B(10)$:	2
Port for node $5 \in B(10)$:	1
Port for node $9 \in B(10)$:	1
Number of red node in $B(3)$ (same clr.):	2
Address of node 3 (same clr.) in T_2:	...
Number of red node in $B(4)$ (same clr.):	6
Address of node 4 (same clr.) in T_6:	...

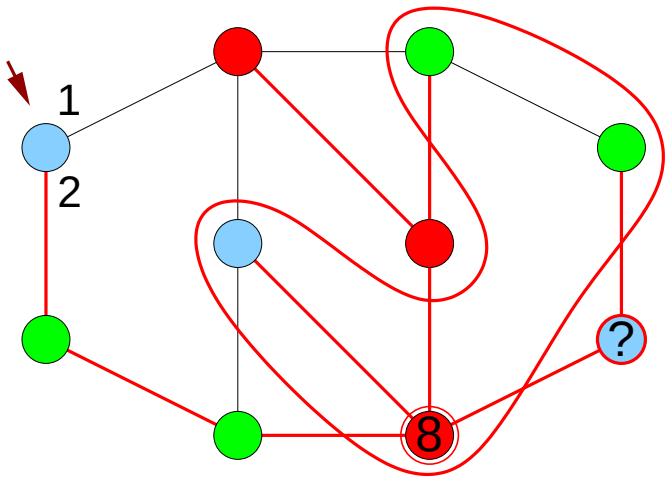
Routing Data Packages



7
DATA

Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

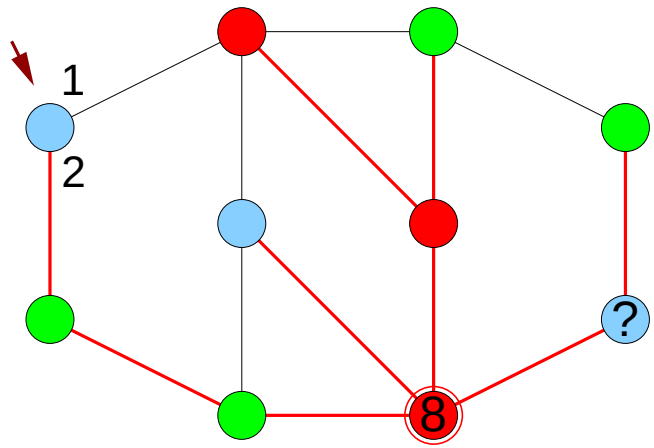
Routing Data Packages



7
DATA

Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

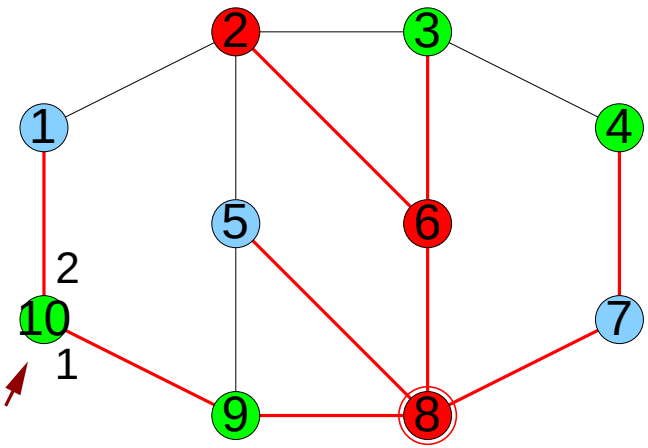
Routing Data Packages



Addr. of 7 in T_8
DATA

Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
Local memory of node 1 for labeled routing in T_6:	
...	...
...	...
Local memory of node 1 for labeled routing in T_8:	
...	...
...	...
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

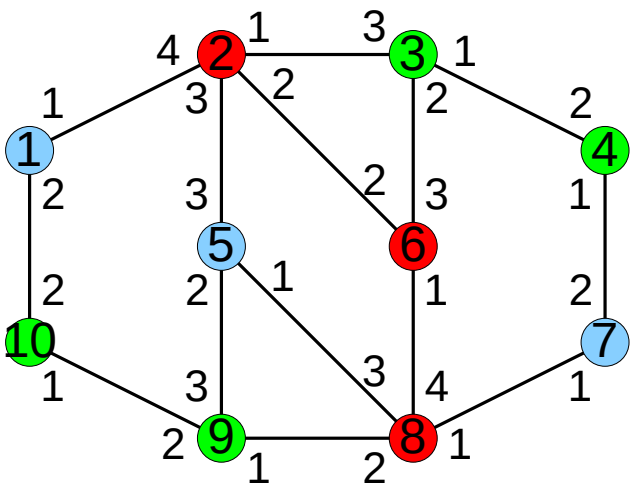
Routing Data Packages



Addr. of 7 in T_8
DATA

Local memory of node 10:	
Own address:	10
Local memory of node 10 for labeled routing in T_2:	
...	...
...	...
Local memory of node 10 for labeled routing in T_6:	
...	...
...	...
Local memory of node 10 for labeled routing in T_8:	
...	...
...	...
Port for node $1 \in B(10)$:	2
Port for node $2 \in B(10)$:	2
Port for node $5 \in B(10)$:	1
Port for node $9 \in B(10)$:	1
Number of red node in $B(3)$ (same clr.):	2
Address of node 3 (same clr.) in T_2:	...
Number of red node in $B(4)$ (same clr.):	6
Address of node 4 (same clr.) in T_6:	...

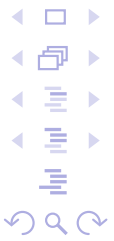
Properties of the Routing Strategy



Local memory of node 1:	
Own address:	1
Local memory of node 1 for labeled routing in T_2:	
...	...
...	...
...	
Port for node 2 $\in B(1)$:	1
Port for node 3 $\in B(1)$:	1
Port for node 5 $\in B(1)$:	1
Port for node 10 $\in B(1)$:	2
Number of red node in $B(7)$ (same clr.):	8
Address of node 7 (same clr.) in T_8:	...

- Header size:
  $O((\log n)^2 / \log\log n$ bits
- Local memory:
  $O(\sqrt{n}(\log n)^3 / \log\log n$ bits
- Stretch 3

Outline



Conclusion

- Low memory consumption even in the case of name-independent routing.
- Applicable to Sensor or Ad-hoc networks?
  - Lower bounds.
  - Is it reasonable to optimize only stretch and local memory consumption?
  - Changing topology: How to update the local data structures?