Quality of a routing scheme

- Routing schemes are judged by...
  - speed of end-to-end delivery
  - speed of reconfiguration/adaptation
  - ratio of control traffic to data traffic
  - *routing-table space required at each host*
- Reducing space at hosts is called *compact routing*
- Reduced space decreases delivery speed (e.g., uses suboptimal paths) and/or increases traffic
  - extreme examples: flooding and random routing
  - one measure: *stretch* (path length / optimal length)

Size of the routing table

Destination	Channel
v_1	c_1
v_2	c_2
v_3	c_1
v_4	c_1
v_5	c_3

table length: n

Channel	Destination
c_1	v_1, v_3, v_4
c_2	v_2
c_3	v_5

table length: degree d

- What if we invert the table?

- So far, forwarding treated as a simple table lookup indexed by destination

Size of the routing table

Destination	Channel
v_1	c_1
v_2	c_2
v_3	c_1
v_4	c_1
v_5	c_3

Channel	Destination
c_1	v_1, v_3, v_4
c_2	v_2
c_3	v_5

savings depends on how well we can compact the destination addresses in this column

How would multicast affect the size of routing tables?

yet, we must still be able to access the table by destination address

min. size: $n(\log_2 n + \log_2 d)$ bits

min. size: $d(\log_2 d + ??)$ bits

Some common compaction techniques

- Tree labeling

  assumes a tree network topology

- Interval labeling

  extension of tree labeling to non-tree network topologies

- Prefix routing

  treats labels (i.e., addresses) as strings and routes using (address) prefixes

Santoro and Khatib tree labeling

basic definitions

- Idea: label hosts with integers 0 to n-1 such that the set of destinations for each channel is an *interval*

  let Z_n denote the set {0, 1, …, n-1}, and use arithmetic modulo n (i.e., $n-1+1 \equiv 0$)

- The *cyclic interval* $[a,b)$ in Z_n is defined by

$$[a,b) = \begin{cases} \{a, a+1, \ldots, b-1\} & \text{if } a < b \\ \{0, \ldots, b-1, a, \ldots, n-1\} & \text{if } a \geq b \end{cases}$$

$[a,a) = Z_n$ and complement of $[a,b)$ is $[b,a)$ for $a \neq b$

[a,b) is called *linear* if $a < b$

Tree labeling, formally

- Theorem

  the nodes of a tree T can be numbered in such a way that for each outgoing channel of each host the set of destinations that must be routed via that channel is a cyclic interval

- Proof

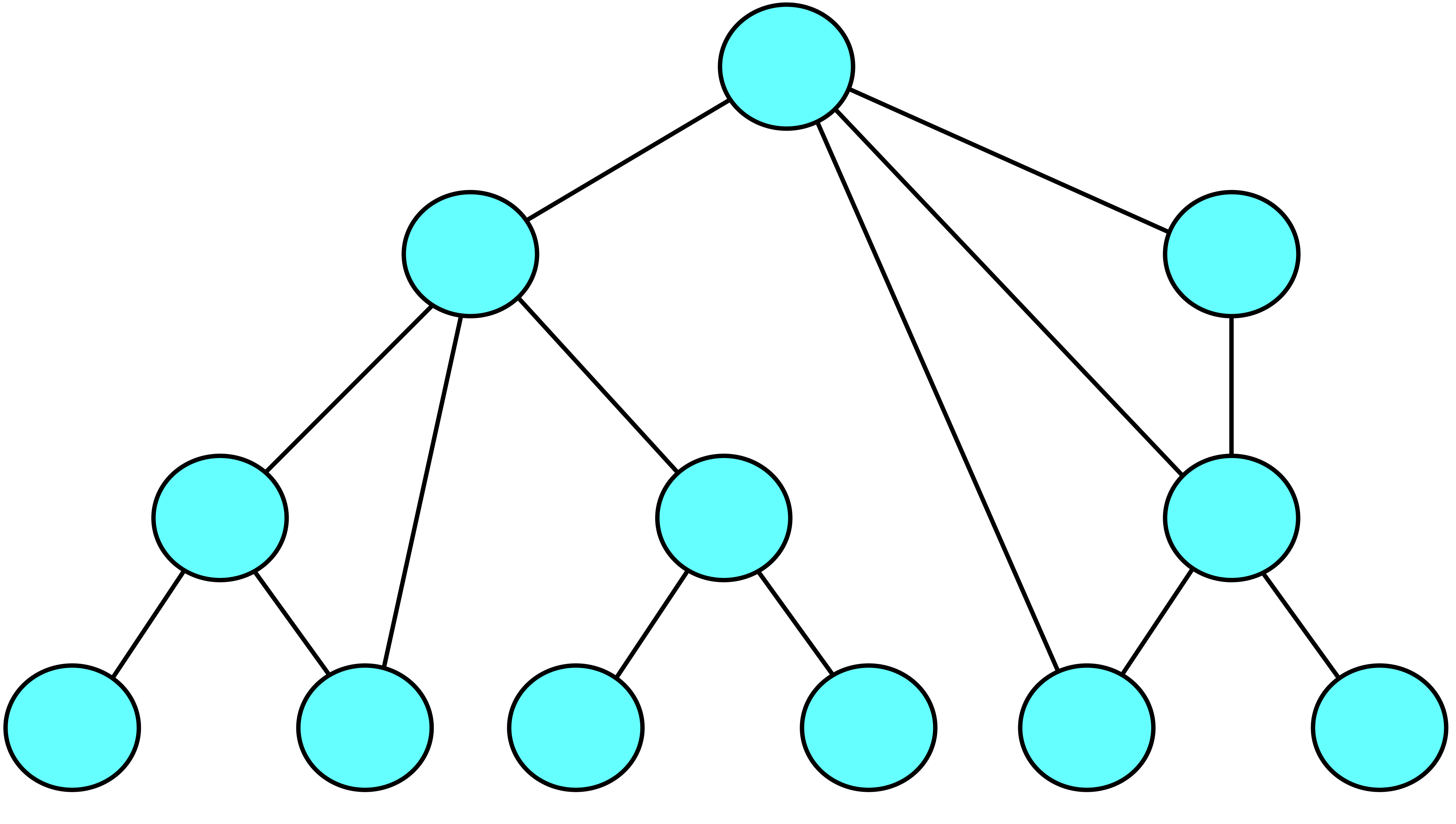
  pick an arbitrary node v_0 as the root and for each w let $T[w]$ denote the subtree of T rooted at w

  number the nodes using a *preorder traversal*, resulting in $T[w]$ having a linear interval labeling

$$[\lambda_w, \lambda_w + |T[w]|)$$

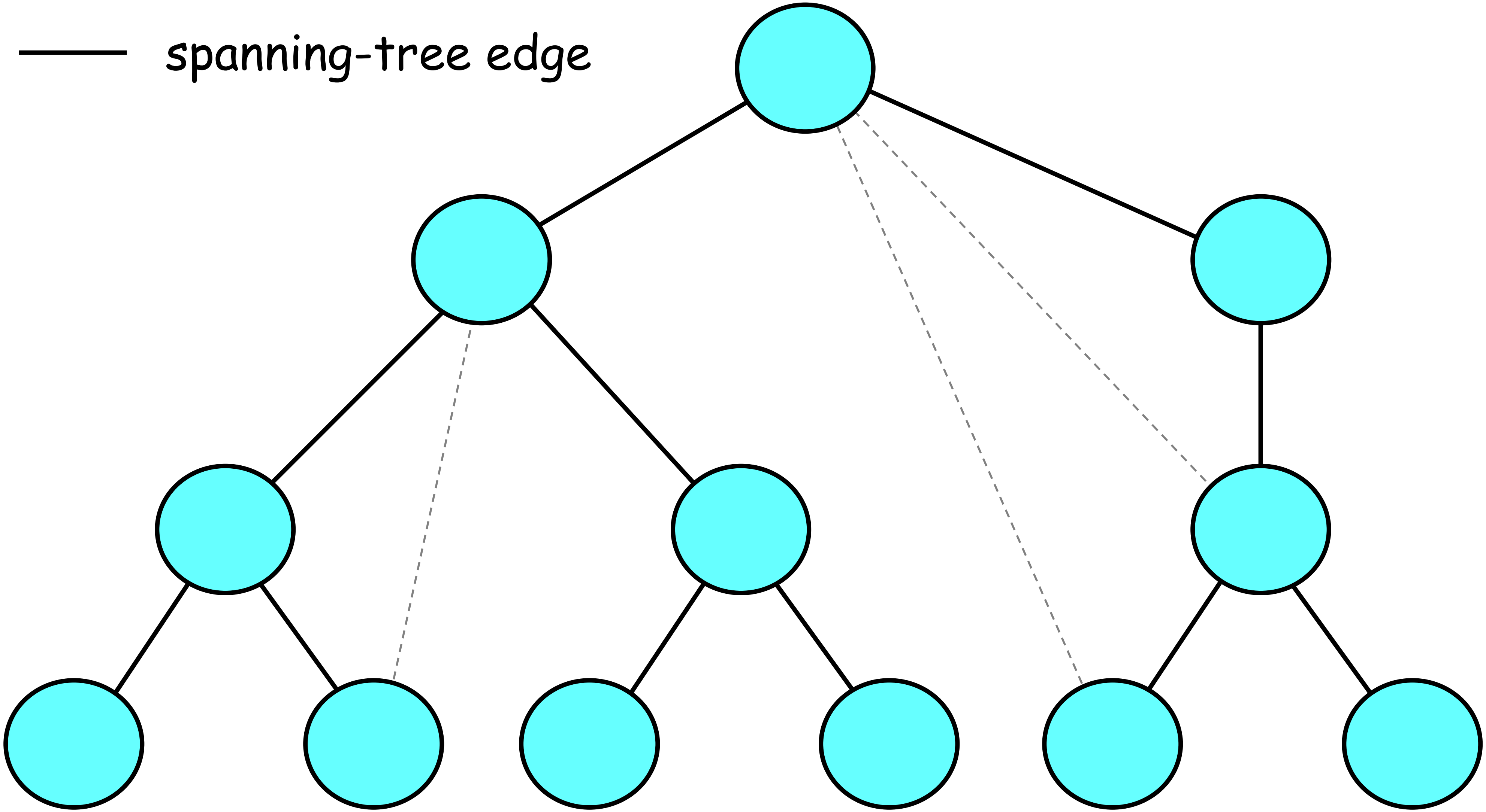
Example tree labeling

raw connectivity graph



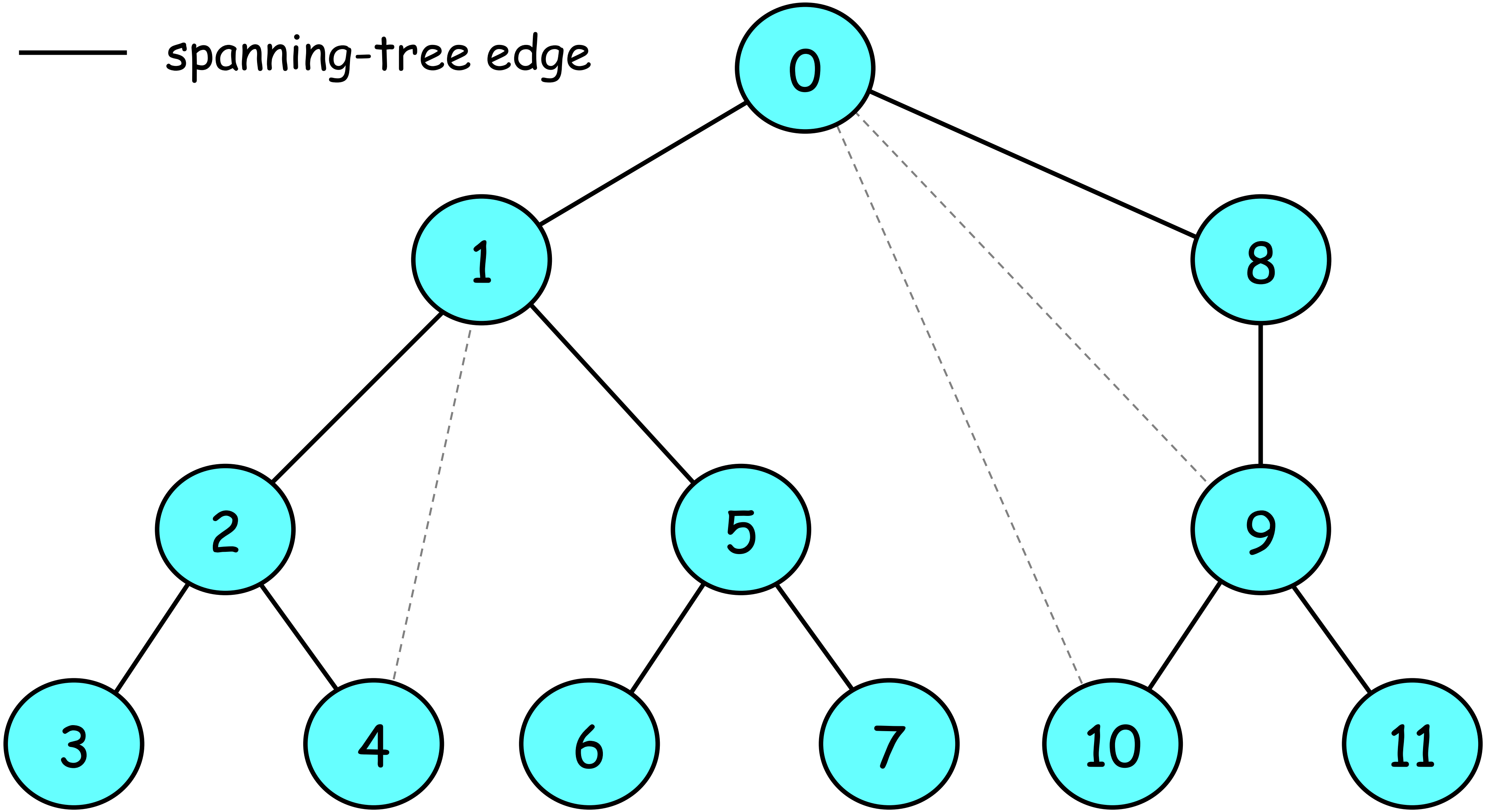
Example tree labeling

superimposed spanning tree

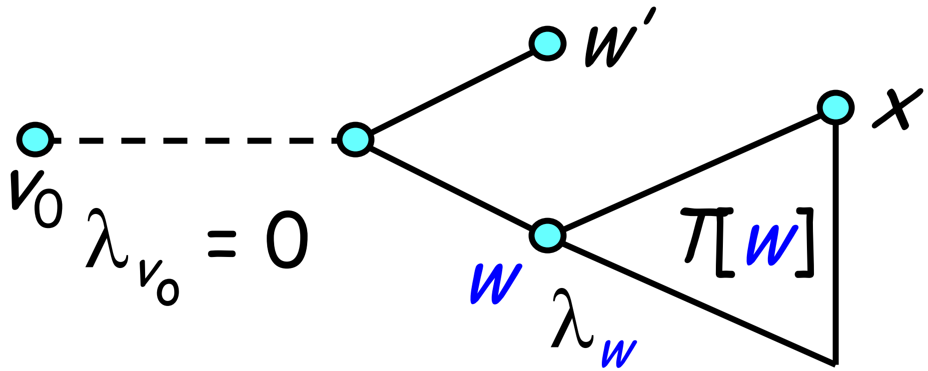


Example tree labeling

preorder traversal labeling



Tree labeling, formally



$\lambda_{w'} = \lambda_w + |T[w]|$

$\lambda_x = \lambda_w + |T[w]|-1$

let $[a_w, b_w)$ denote the interval of numbers assigned to the nodes in $T[w]$

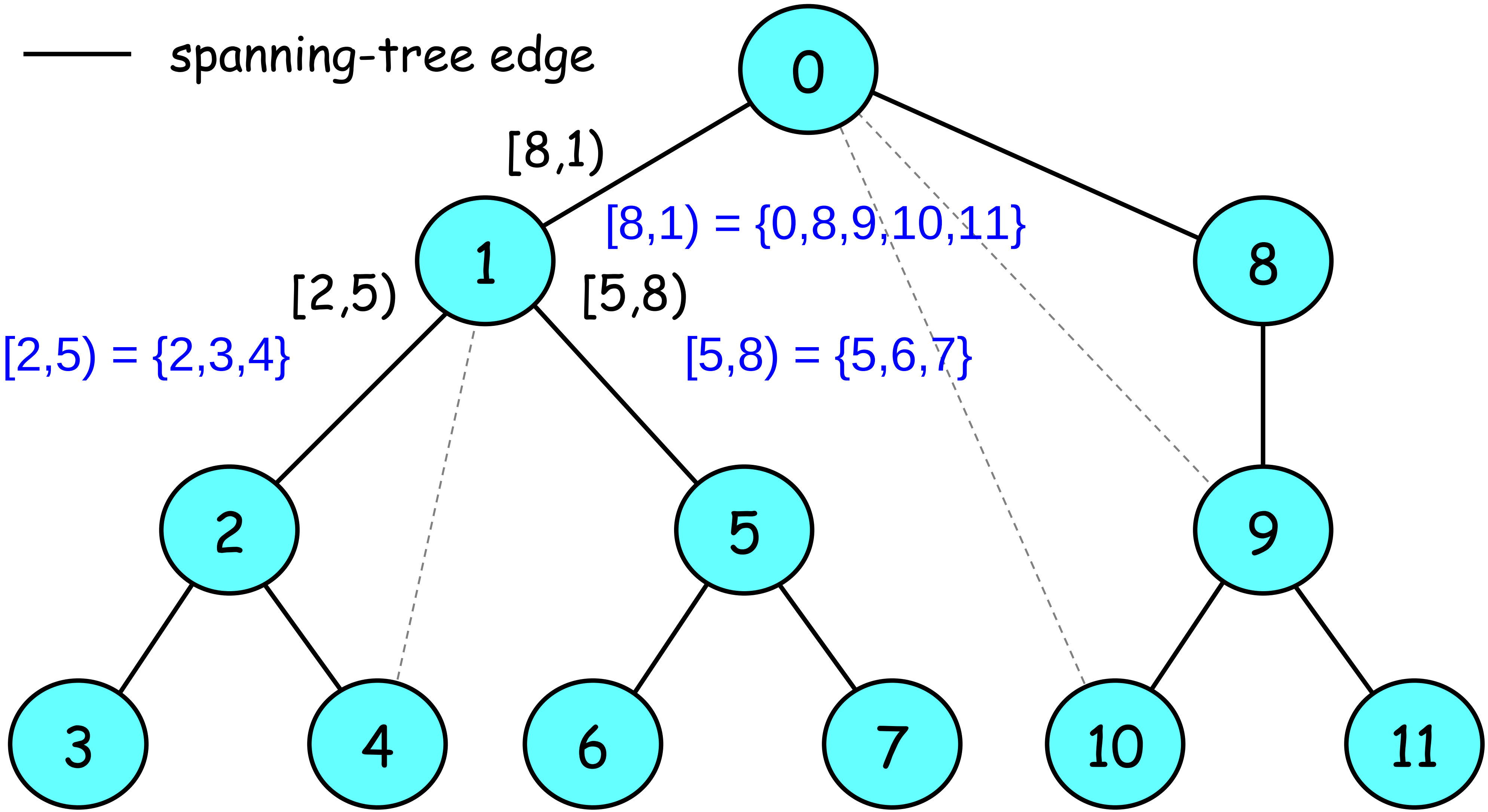
a neighbor of w is either a child or the parent of w

w forwards to a child u the messages with destinations in $T[u]$, i.e., the nodes with numbers in $[a_u, b_u)$

w forwards to its parent the messages with destinations not in $T[w]$, i.e., the nodes with numbers in $Z_n \setminus [a_w, b_w) = [b_w, a_w)$

Example tree labeling

cyclic intervals at node 1



Tree labeling

size of the routing table

- *A* single cyclic interval can be represented using $2\,log_2 n$ bits by giving the start and end points
- Used in routing tables, the intervals are disjoint with union Z_n, so $\log_2 n$ bits are sufficient

  only the start point needs to be stored

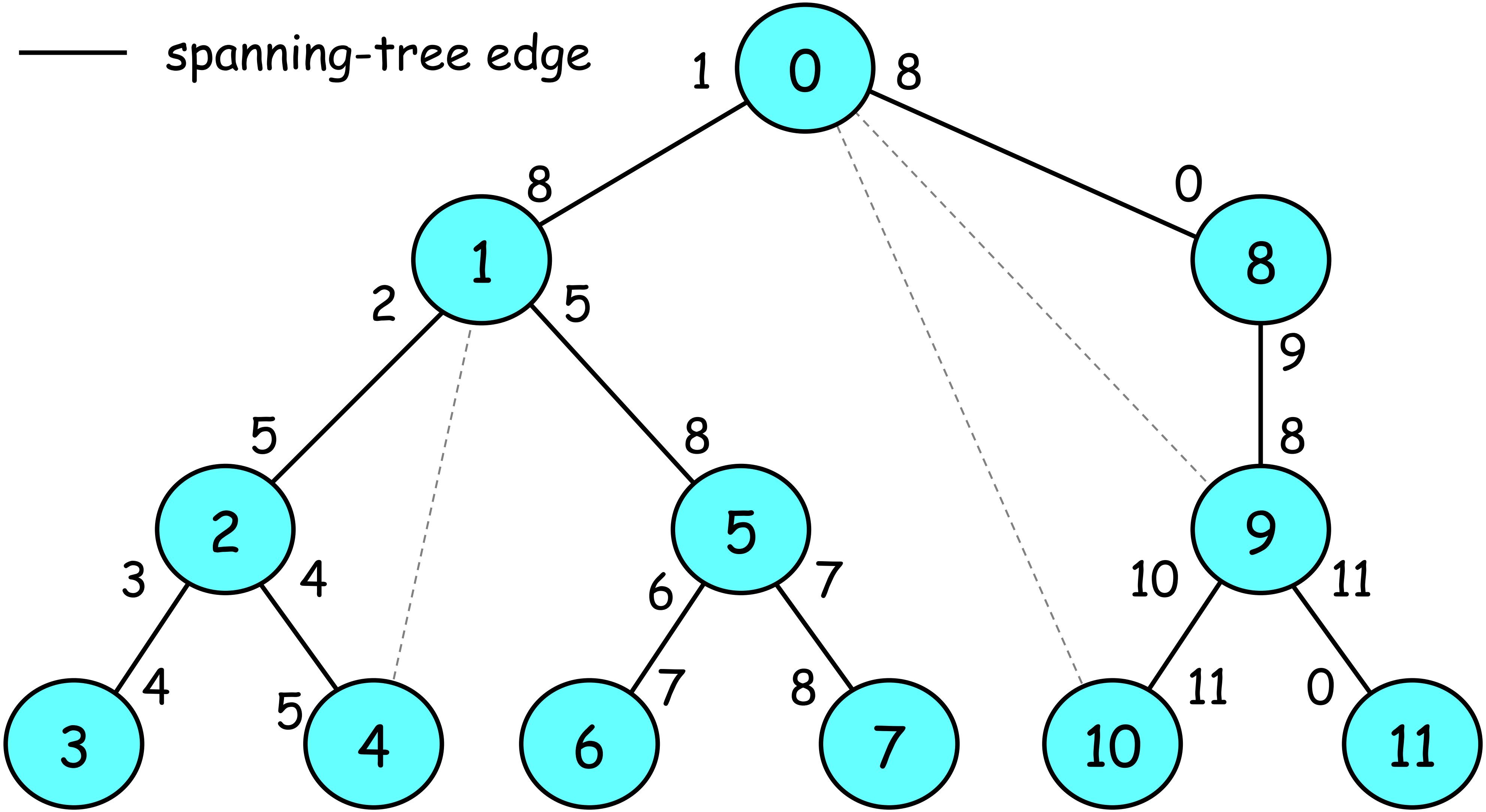
  end point is the start point of the next interval

  start point of interval for channel *uw* at *u* is given by

$$\alpha_{uw} = \begin{cases} \lambda_w & \text{if } w \text{ is a child of } u \\ \lambda_u + |T[u]| & \text{if } w \text{ is the parent of } u \end{cases}$$

Example tree labeling

compact labeling



Tree labeling

forwarding

- Assume u is of degree deg_u and the channels are labeled with $\alpha_1, \ldots, \alpha_{deg_u}$, where $\alpha_1 < \cdots < \alpha_{deg_u}$

```
% message m with destination address d received at u
if d = λu then
  deliver m
else
  select αi such that d ∈ [αi,αi+1)
  send m via channel αi
```

- Channel labels partition Z_n into deg_u segments, one per channel; at most one is non-linear interval
- If the labels are sorted, then label can be found in $O(\log_2 deg_u)$ steps

Tree labeling

observations

- Channels not belonging to T are not used

  $\Rightarrow$ waste of network resources

- Traffic is concentrated within in the spanning tree

  $\Rightarrow$ congestion

- Single failures of a channel in T cause major problems

  $\Rightarrow$ partitioning of the network

Van Leeuwen and Tan interval labeling

overview

- Extension of tree labeling to non-tree networks such that (almost) every channel is used
- Basic structure is the spanning tree T

  a *frond edge* is an edge not in T

  v is an *ancestor* of u if $u \in T[v]$

  tree edges labeled as in the tree-labeling technique

  problem is to label frond edges and choose T

- There exists a T such that all frond edges are between a node and an ancestor of that node

  obtained by a depth-first traversal of the network

Interval labeling

construction rules using depth-first traversal

- Labels assigned as in a preorder traversal of T

  subtree $T[w]$ labeled with numbers in $[\lambda_w, \lambda_w + |T[w]|)$

  let $k_w = \lambda_w + |T[w]|$

- Let label of edge uw at u be called α_{uw}

  (1) if uw is a frond edge then $\alpha_{uw} = \lambda_w$

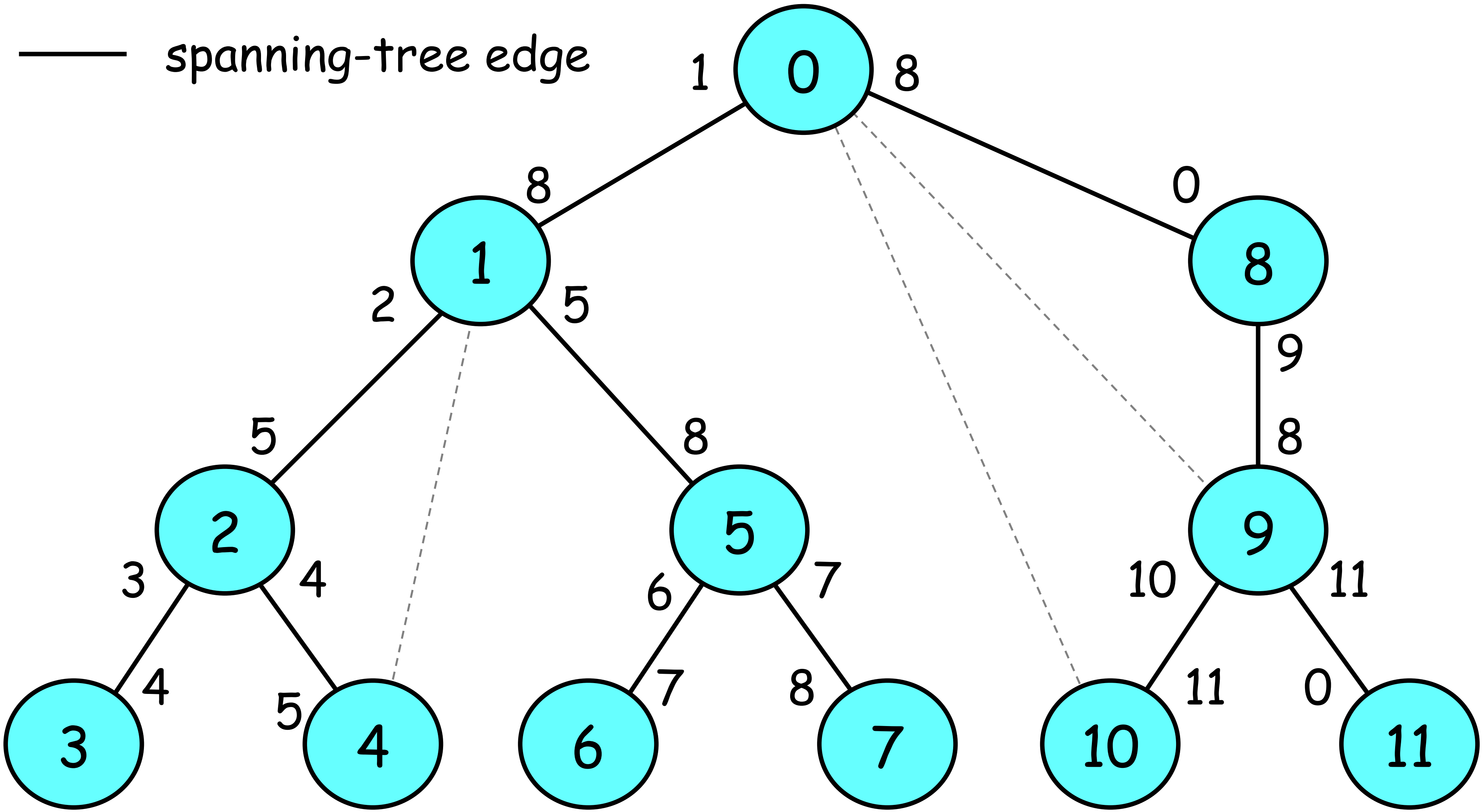
  (2) if w is a child of u (in T) then $\alpha_{uw} = \lambda_w$

  (3) if w is the parent of u then $\alpha_{uw} = k_u$ unless $k_u = n$ and u has a frond to the root

  (4) if w is the parent of u, u has a frond to the root, and $k_u = n$ then $\alpha_{uw} = \lambda_w$

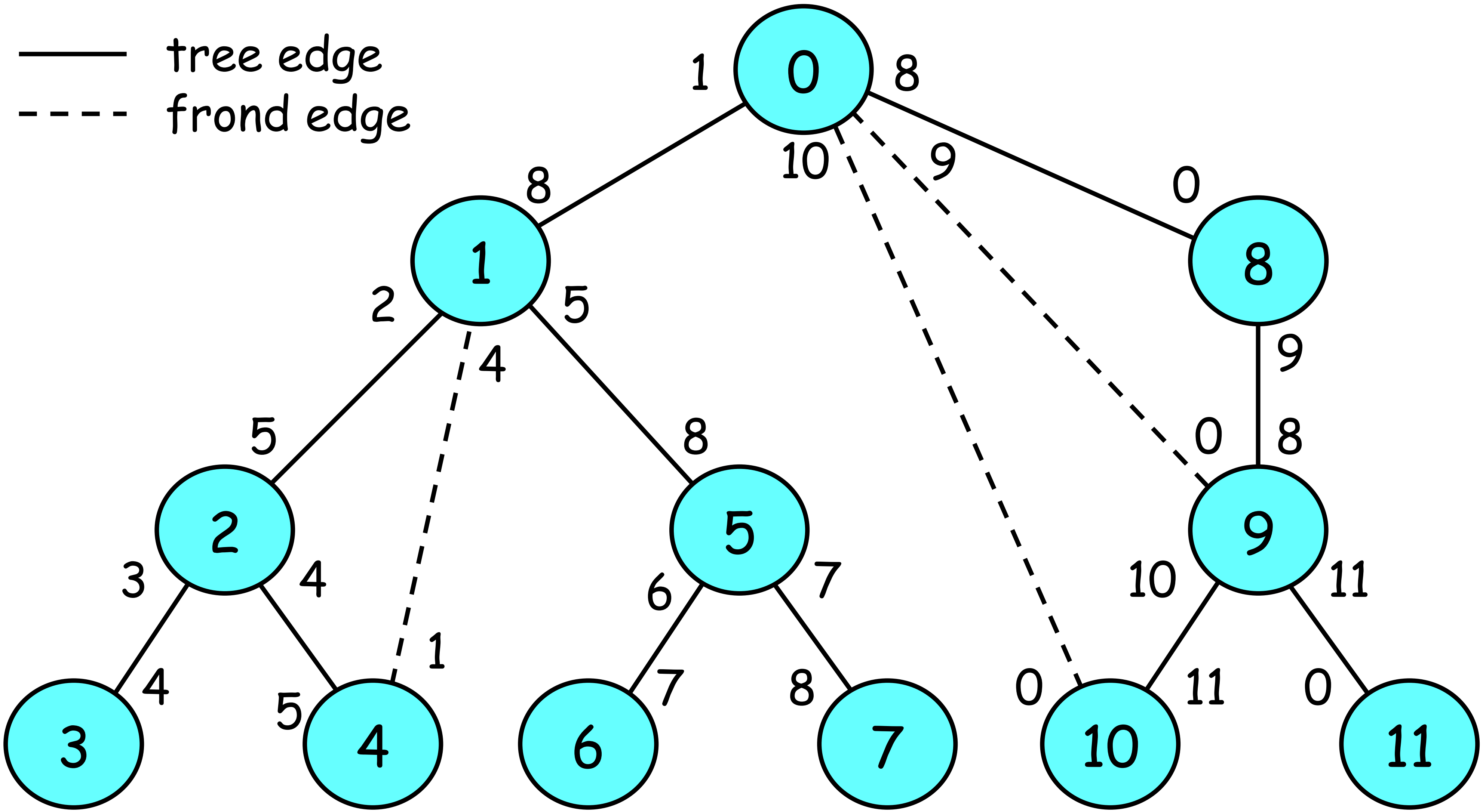
Example interval labeling

basic tree labeling



Example interval labeling

making use of fronds



- Forwarding is done as for tree-labeling technique

Interval labeling
observations

- Poor robustness

  not possible to adapt a depth-first traversal labeling if channel or host added/removed from the network

  difficult to preserve the “frond” property

  minor modification in topology may require complete recomputation of routing tables, including assignment of new labels (i.e., addresses) to each host

- Non optimality

  depth-first traversal labeling may route messages via paths of length $\Omega(n)$, even for small-diameter networks

Bakker et al. prefix routing

- Routing tables can be computed using arbitrary spanning trees, increasing robustness

  if a channel is added between two existing hosts then the new channel becomes a frond

  if a new host, and therefore one or more new channels, is added then one of the channels is used to extend the spanning tree and the others become fronds

  and efficiency

  free to choose a small-depth spanning tree

Bakker et al. prefix routing

sketch

- Host and channel labels are *strings* rather than integers
- To select a channel, the forwarding algorithm considers all channels that are a *prefix* of the destination address, and selects the *longest* such prefix
- Example

  assume a host has channels: **aabb**, **abba**, **aab**, **aabc**, and **aa**, and must forward a message with destination address **aabbc**

  which channel is chosen?