Chapter 7: Naming & Addressing

Goals of this chapter

- This short chapter looks at non-standard options for denoting the senders/receivers of messages
  - Traditional (fixed, wireless, ad hoc): Denote individual nodes by their identity
  - WSN: *Content-based* addresses can be a good complement
- When addresses are not given a priori, they have to be determined "in the field"
  - Some algorithms are discussed

Names vs. addresses(1)

- ***Name:*** Denote/refer to "things"
  - Nodes, networks, data, transactions, …
  - Often, but not always, unique (globally, network-wide, locally)
  - Ad hoc: nodes – WSN: Data!

- ***Addresses:*** Information needed to ***find*** these things
  - Street address, IP address, MAC address
  - Often, but not always, unique (globally, network-wide, locally)
  - Addresses often hierarchical, because of their intended use in, e.g., routing protocols

Names vs. addresses(2)

- Services to map between names and addresses
  - E.g., DNS (mapping user-friendly names to the IP address)
- Sometimes, same data serves as name and address
  - IP addresses are prominent examples

Use of address and names in sensor networks (1)

- Unique node identifier (UID):
  - combination of
    - Vendor name,
    - a product name
    - a serial number
    - assigned at manufacturing time

Use of address and names in sensor networks (2)

- MAC address:
  - used to distinguish between one-hop neighbors of a node
  - Application:
    - Contention-based MAC protocol: unicast
    - Overhearing avoidance for energy saving

Overhearing

- Overhear
  - In a unicast: one source, one destination
    - Wireless medium is a broadcast medium,
    - All the source's neighbors that are in receive state
    - Hear a packet and drop it when it is not destined to them

Transceiver states

- Transceivers can be put into different operational ***states***, typically:
  - ***Transmit***
  - ***Receive***
  - ***Idle*** – ready to receive, but not doing so
  - ***Sleep*** – significant parts of the transceiver are switched off
    - Not able to immediately receive something
    - ***Recovery time*** and ***startup energy*** to leave sleep state can be significant

Use of address and names in sensor networks (3)

- Network address:
  - Denote over multiple hops
  - Application:
    - routing

Use of address and names in sensor networks (4)

- Network Identifiers:
  - Geographically overlapping wireless networks
  - Work in the same frequency band
  - Application:
    - Sensor data belonging to different patient

Use of address and names in sensor networks (5)

- Resource Identifiers:
  - A name
  - Application:
    - www.xemacs.org

Use of address and names in sensor networks (6)

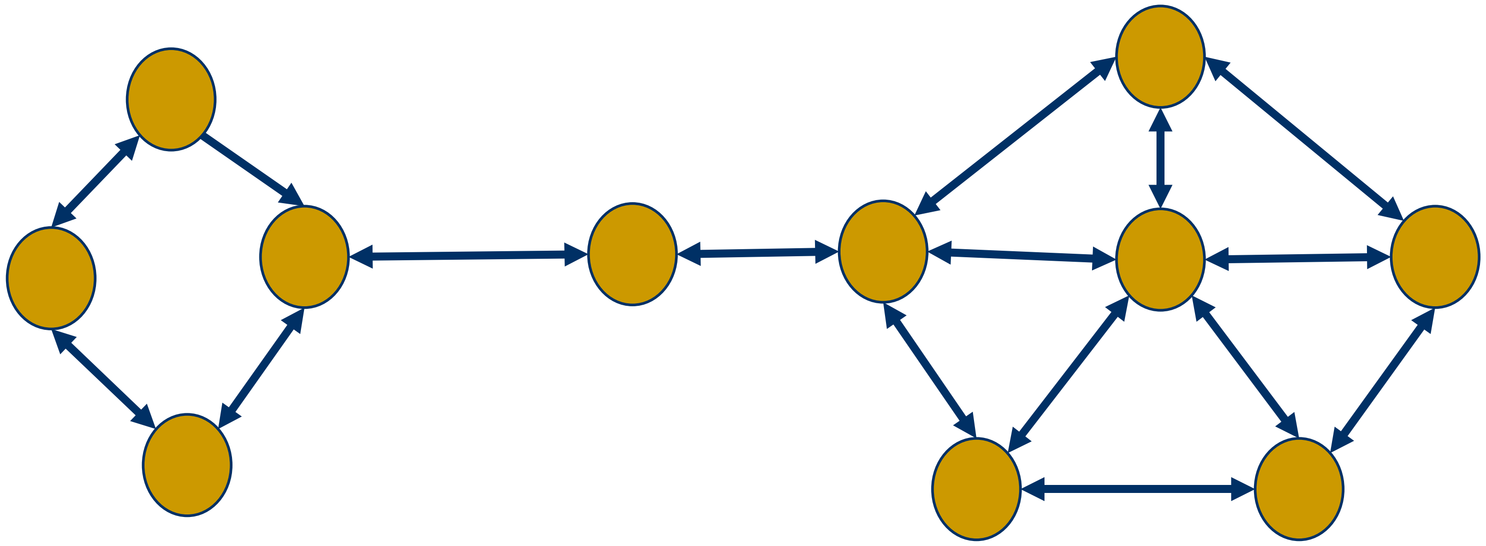
- Binding
  - Map between addresses used by different protocol layers
  - E.g., IP addresses are bound to MAC address by ARP (Address Resolution Protocol)

Issues in address management(1)

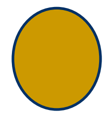
- **Address allocation**:
  - Assign an address to an entity from a given pool of possible addresses
  - Distributed address assignment (centralized like DHCP(Dynamic Host Configuration Protocol) does not scale)

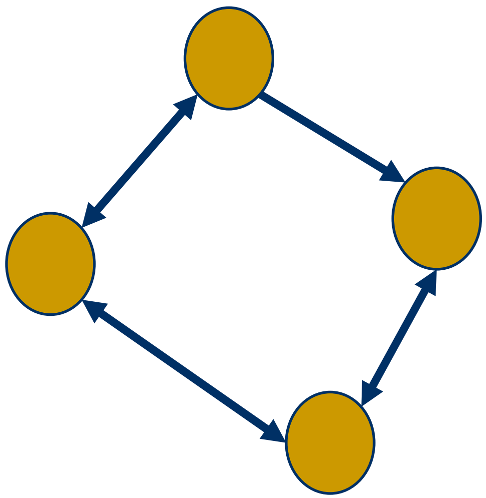
Issues in address management(2)

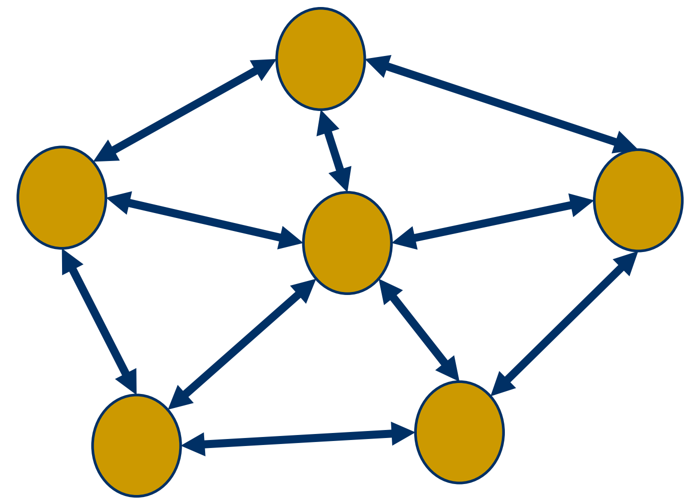
- Address deallocation:
  - Once address no longer used, put it back into the address pool
  - Because of limited pool size
  - Graceful deallocation: a node explicitly send out control packet to give up its address
  - Abrupt deallocation: the node disappears or crashes



Deallocation or not?







Issues in address management(3)

- Address representation
  - A format for representing addresses

Issues in address management(4)

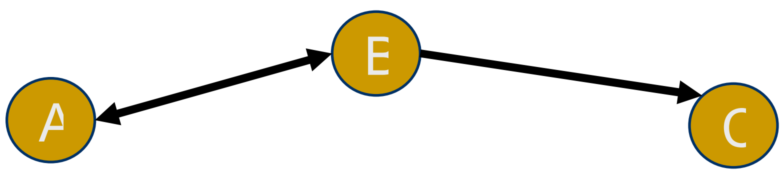
- Conflict detection & resolution (***D**uplicate **A**ddress **D**etection*)
  - What to do when the same address is assigned multiple times?
  - Can happen e.g. when two networks merge

Uniqueness of address(1)

- Globally unique:
  - IEEE MAC address used in Ethernet and Token Ring
- Network wide unique:
  - a and b can't communicate if $a \in A$ and $b \in B$, with $A <> B$

Uniqueness of address(2)

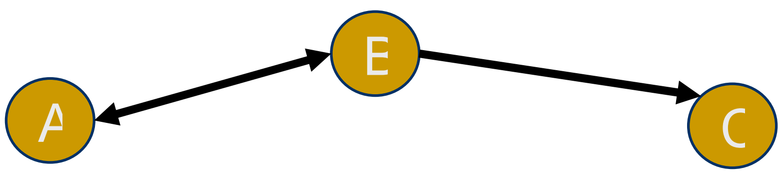
- Locally unique:
  - For MAC address, they are unique only within a two-hop neighborhood



  - A and C can't have the same address

Uniqueness of address(3)

- Locally unique:
  - For MAC address, they are unique only within a two-hop neighborhood



    - A and C can't have the same address

  - No two same type sensor nodes have the same address.

Address allocation and assignment

- Priori:
  - During manufacturing process…
- On demand
  - By using an address assignment protocol

On demand address assignment

- Centralized
  - One single authority/node taking care of the address pool
  - When network partition, …
- Distributed
  - All nodes play the same role in address assignment

Problems with centralized scheme

- Centralized solution do not scale well
  - Significant traffic
- Not reachable when network partition
- Periodically to detect abrupt deallocation

Distributed address assignment (1)

- Not always possible to guarantee network wide uniqueness
- Solution:
  - Live with few address conflicts
  - Detect conflicts and resolve

Duplicate address detection(1)

- Strong DAD (duplicate address detection)

If node a (b) assigned address at time t_0 (t_1),

Then,

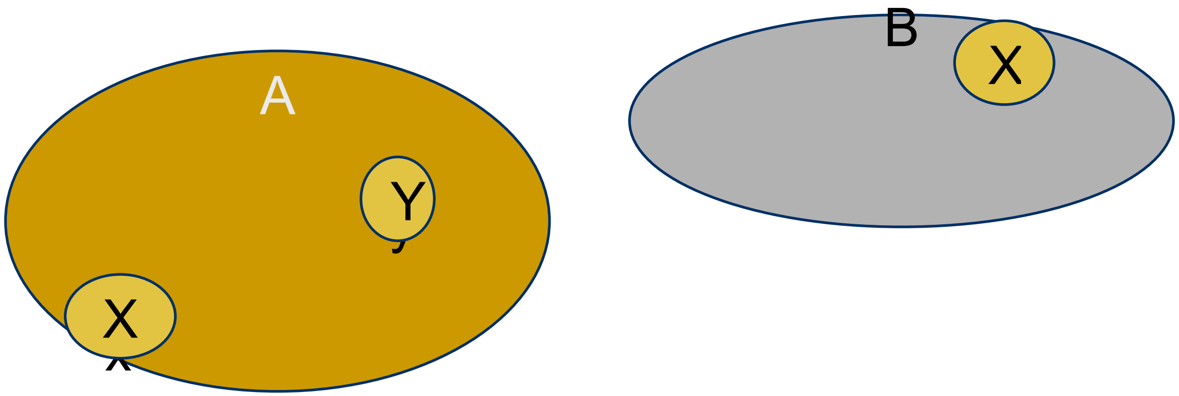
this duplicate assignment must be detected

latest at time $t_1 + T$,

where T is some fixed time bound

Duplicate address detection(2)

- Weak DAD (duplicate address detection)
  - Duplicate addresses are tolerated as long as they do not distort ongoing session
  - E.g.,



Distributed address assignment(2)

- approach 1: (Naïve approach)

  random address assignment

Conflict-free assignment probability:

$$P(n,k) = 1.\frac{n-1}{n}\cdots\frac{n-k+1}{n} = \frac{k!}{n^k}.\binom{n}{k}$$

Distributed address assignment(3)

- Naïve approach: random address assignment

$$P(n,k) = 1.\frac{n-1}{n}\ldots\frac{n-k+1}{n} = \frac{k!}{n^k}.\binom{n}{k}$$

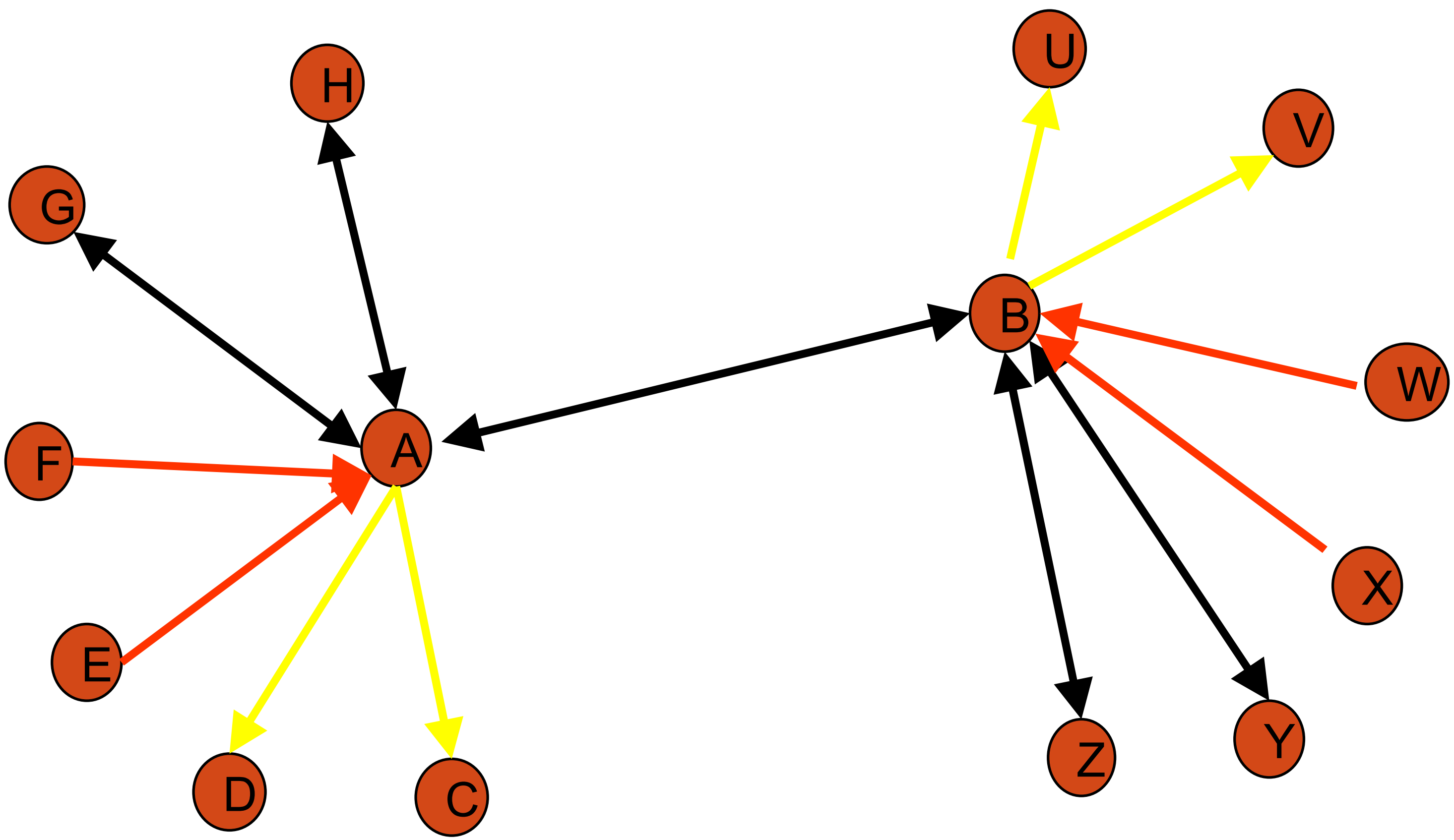
- $N = 2^{14} = 16384$

Node #	Probability without conflict
50	0.9
100	0.72
150	0.5
200	0.3
250	0.15

Distributed address assignment(4)

- Approach 2: Avoid addresses used in local neighborhood

Distributed address assignment(5)



Distributed address assignment(6)

- Approach 3: Repair any observed conflicts

Distributed address assignment(7)

- Example 1 of approach 3:
  - Each node carry a temporary address and a proposed fixed address
  - Send an address request:
  - Try to find a path to a node having the same fixed address
  - If there exists such a node, an address reply packet is generated and sent back to temporary address
  - Delay bound

Distributed address assignment(8)

- Example 2 of approach 3:
  - Initiator: already have an address
    - keep a table of all known address assignment
  - Requester contacts a neighboring initiator
    - Initiator Pick an unused address
    - Initiator disseminates the proposed new address to all nodes in network
    - Each node receive the message check its own table

Distributed address assignment(9)

- Other approach:
  - Hierarchical address auto-configuration algorithm for IPv6 intended for MANETs
    - Some nodes in the network become leader nodes and assign addresses to other nodes
  - Clustering

Addressing Overhead

- The frequency with which address are used
- The size of their representation

Address selection and representation

- Address selection is greedy.
  - Select the lowest possible nonconflicting address
  - Lower address have a higher relative frequency and have a nonuniform address distribution

Length of address is not uniform

Relative frequencies of MAC address under the greedy algorithm

Frequen-cy	Address						
Density	0	10	20	30	40	50	60
4	0.2	0.01	0	0	0	0	0
6	0.12	0.04	0	0	0	0	0
10	0.065	0.05	0.01	0	0	0	0
15	0.04	0.038	0.03	0.01	0	0	0
30	0.02	0.019	0.017	0.015	0.013	0.01	0

Length of address is not uniform

Non-uniform address assignment is useful

when

we can determine the relative frequencies/address distribution of MAC address

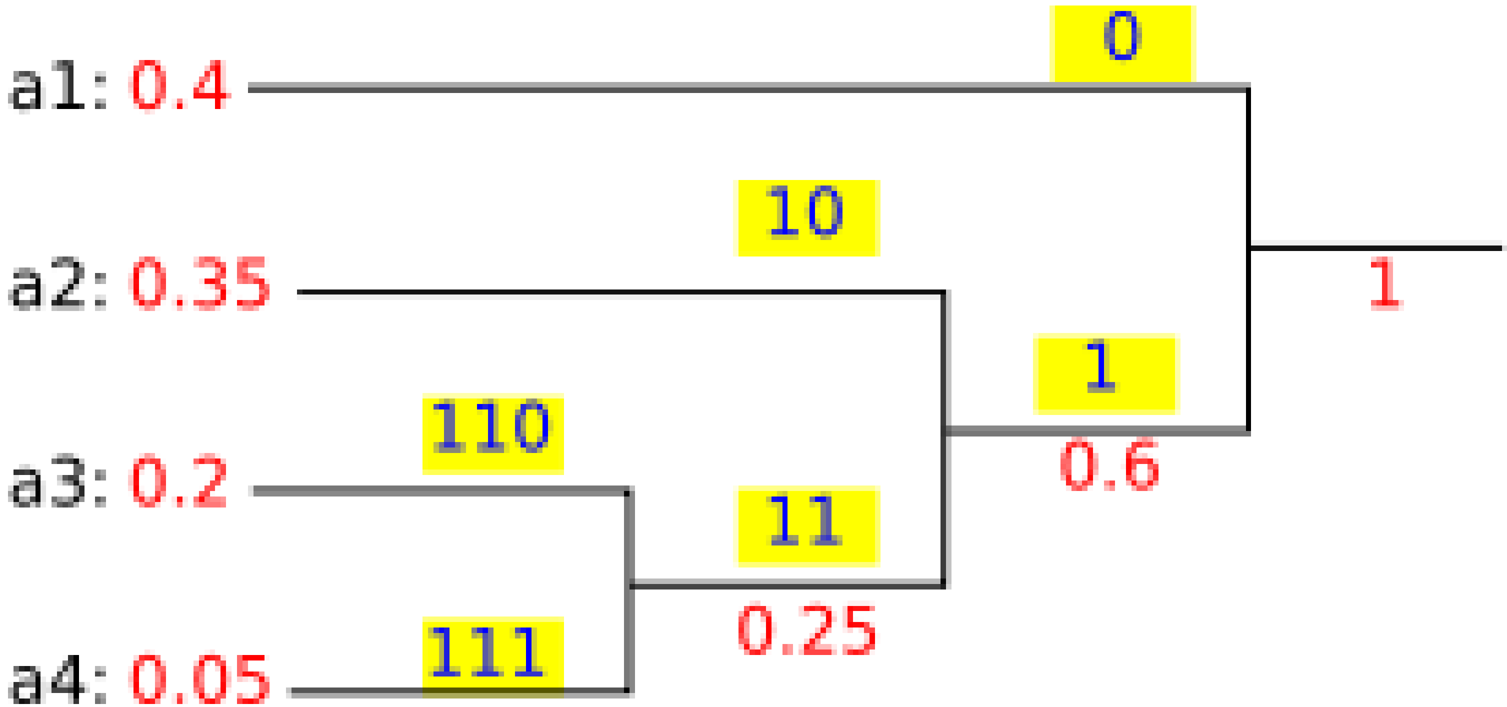
Length of address is not uniform

To implement non-uniform address assignment,

The receiver has to parse the variable-length address and find the remaining packet.

Huffman coding

- Each code can be uniquely decoded
- To accomplish this, Huffman coding creates what is called a "Huffman tree", which is a binary tree
  - Prefix free
  - No code word is a prefix of any other code word.



a1: 0.4
a2: 0.35
a3: 0.2
a4: 0.05
0
10
1
110
11
111
0.6
0.25
1

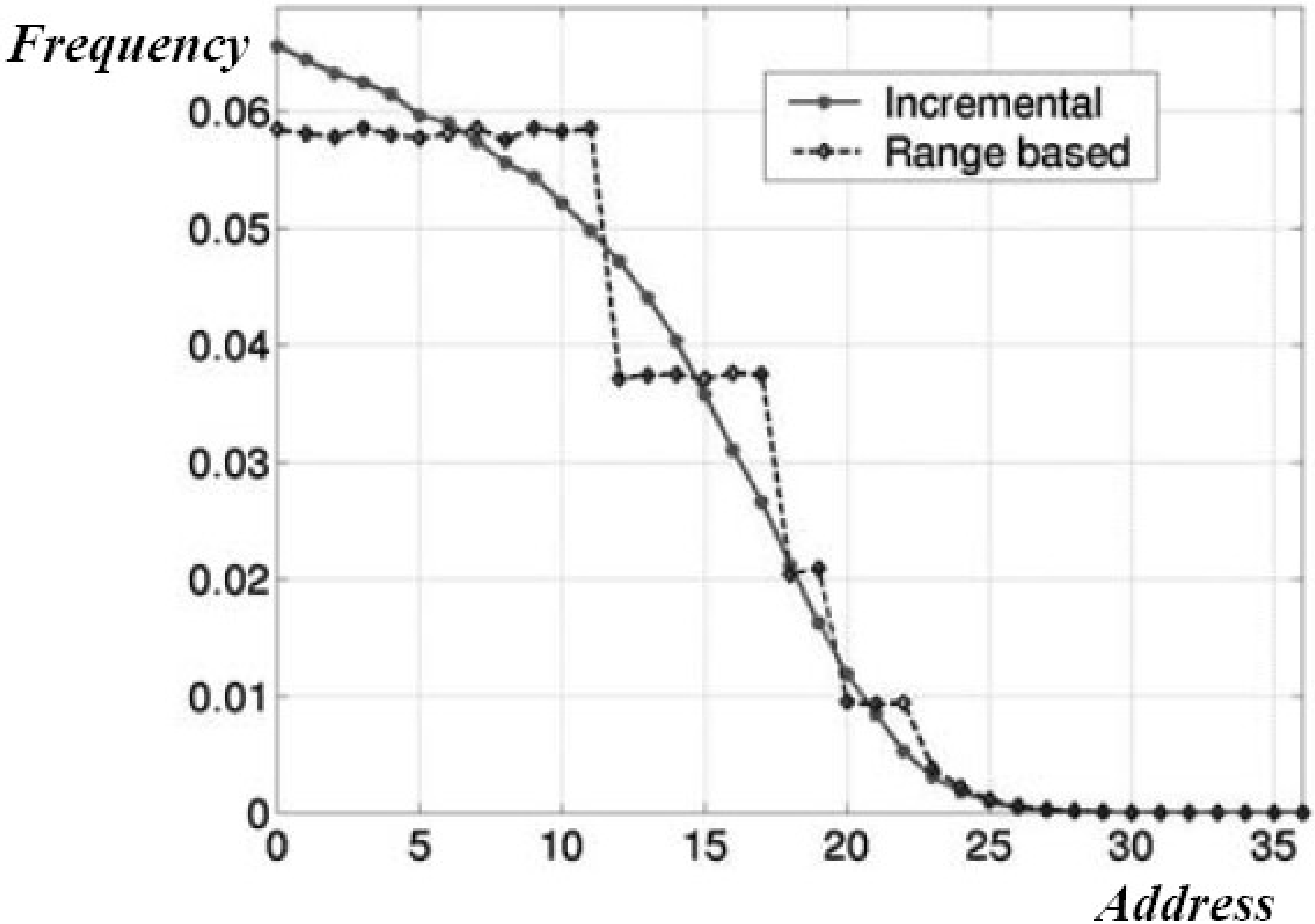


Figure 3: Example of address selection frequency

Frequency	Address						
Strategy	0	5	10	15	20	25	30
Lowest first	0.06	0.055	0.048	0.038	0.02	0.005	0
Range based	0.055	0.055	0.055 0.038	0.038	0.022 0.038	0.015 0.008	0

4 bit

5 bit

6 bit 7 bit

8 bit

Addr size	Network size N			
density	10^1	10^2	10^3	10^4
5	2.5	3.4	3.5	3.6
10	3.25	4.25	4.5	4.7
15	3.5	4.85 (5.4)	5.1 (5.6)	5.2 (5.75)
20	3.5	5.2 (5.7)	5.5 (6.1)	5.6 (6.2)

Fixed-length scheme

Content-based addresses

- Recall:
  - Paradigm change from id-centric to data-centric networking in WSN
- Supported by content-based names/addresses
  - Do not described involved nodes (not known anyway), but the ***content*** itself the interaction is about

Content-based addressing: Describe *interests*

- ***Interests*** describe relevant data/event
  - Used, e.g., by directed diffusion (see later chapter)
  - Nodes match these interests with their locally observed data

Content-based addressing: Describe *interests*

- Format: Attribute-Value-Operation
  - <attribute, value, operation>, e.g.: <TEMP, 20° C, GE>
  - Attributes: temperature, pressure, concentration, …
  - Operations:

Operator name	Meaning
EQ	Matches if actual value is equal to `value`
NE	Matches if actual value is not equal to `value`
LT	Matches if actual value is smaller than `value`
GT	Matches if actual value is greater than `value`
LE	Matches if actual value is smaller or equal to `value`
GE	Matches if actual value is larger or equal to `value`
EQ_ANY	Matches anything, `value` is meaningless
IS	Specifies a literal attribute

Matching algorithm

- Check whether an interest matches the locally available data

```
parameters: attribute sets A and B
   // A corresponds to the interest, B to the data message

foreach attribute a in A where a.op is formal {
  matched = false
  foreach attribute b in B where
          a.key == b.key and b.op is actual {
    if b.val satisfies condition
       expressed by a.key and a.val then {
      matched = true
    }
  }
  if (not matched) then {
    return false
  }
}
return true;    // matching successful!
```

Geographic addressing

- Express addresses by denoting physical position of nodes
  - Can be regarded as a special case of content-based addresses
  - Attributes for x and y coordinates (and maybe z)

Geographic addressing

- Options
  - Single point
  - Circle or sphere centered around given point
  - Rectangle by two corner points
  - Polygon/polytope by list of points
  - …

Conclusion

- Addresses can be assigned distributedly
- Non-id-centric addresses give additional expressiveness, enables new interaction patterns than only using standard addresses
- These addresses have to be supported by specific protocols, in particular, routing protocols