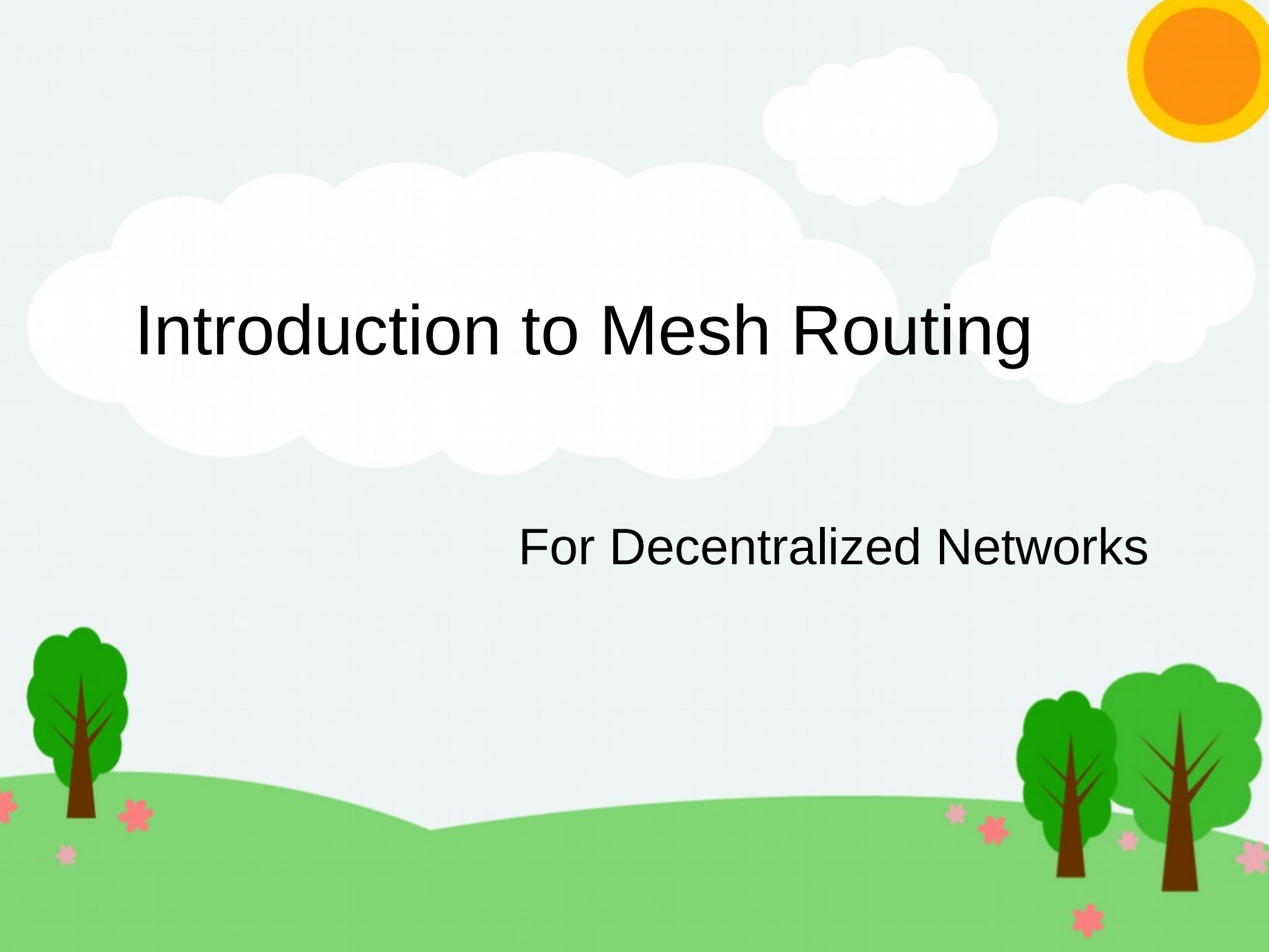
Introduction to Mesh Routing

For Decentralized Networks

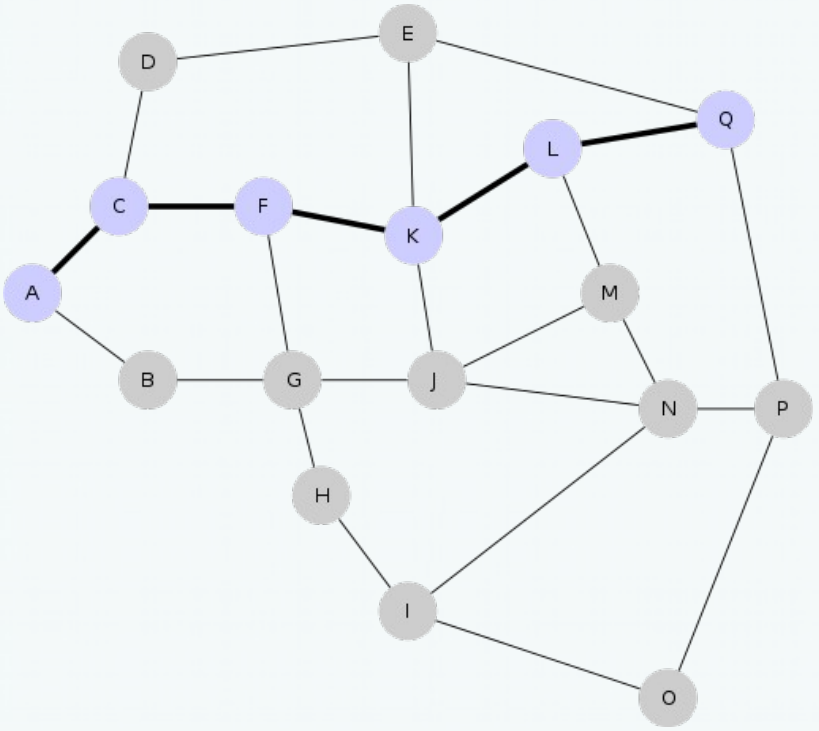


What is routing?

- Getting a (data) packet from A to B
  - Forward decision
  - Needs to be effectively loop free
- Best route decided by metric
  - Hop Count
  - Bandwidth
  - Time
  - Cost

What is a Mesh Network?

- (IP) Network of Computers (nodes)
- arbitrary topology
  - Loops!
- Restricted topology
  - Line
  - Lattice
  - Tree
  - ...



Mesh Types

- Node movement: **Mobile**
  - Usually **wireless** networks
- Nodes appear/disappear: **Ad-hoc**
  - Start with no memory / clean slate
- Virtual
  - Abstract Topology over network
  - Software Defined Networks
- => To clear distinction
  - Depends on task at hand

Mesh Protocols

- **M**obile **A**dhoc **M**esh **N**etworks (MANET)
  - Batman-adv
  - OLSR
  - Babel
  - BMX7
- Non Mobile Mesh Networks
  - BGP (Border Gateway Protocol)
    - Backbone protocol of the Internet
    - Between Internet Service Providers (ISP)
  - OSPF
    - Inside ISPs
    - Uses a spanning tree

Prerequisites

- Network topology (handled on previous slides)
- Path Metric
- Routing schemes
- Distance Vector / Link State
- Proactive / Reactive

Path Metric

- When is a path better than another?
- Routing Metric
  - Hops (RIP, academic protocols)
  - Packet Loss (old batman-adv)
  - Bandwidth (batman-adv, olsr)
  - Energy

Routing Schemes

- Name independent
  - Node identifier is opaque value
  - e.g. MAC address
- Named dependent
  - Node identifier serves as address
  - e.g. (global) IP address

Distance Vector

- e.g. batman-adv, geometric routing
- Common approach:
  - called Table Driven
  - Node maintains table
    - Original sender
    - Received from neighbor
    - Path Cost
  - It’s distributed Dijkstra shortest path algorithm

Link State

- e.g. OLSR
- Whole topology in memory
  - Allows whole network path planning
  - Allows Policy routing
- Hand in Hand with “Source Routing”
  - header of packet contains whole path
  - Pro: More control over the path to take
    - Policy Routing!
  - Cons: Inflexible in case of topology changes

Proaktiv/Reactive Routing

- Why send all the management data when no user data needs to be routed?
- Reactive
  - Discover routing information when needed
  - Useful for low traffic networks
- Proactive
  - Keep all routing data up to data
  - Efficient when there is much user data

The Routing Task

- Imagine
  - You are a node that wakes up
  - No knowledge of the surrounding
  - Other nodes are in the same situation
  - You have a unique name (e.g. MAC address)
  - You get a packet for somebody
- What to do?
  - Discover neighbors
  - Forward packet -> which neighbor

Naiive Approach

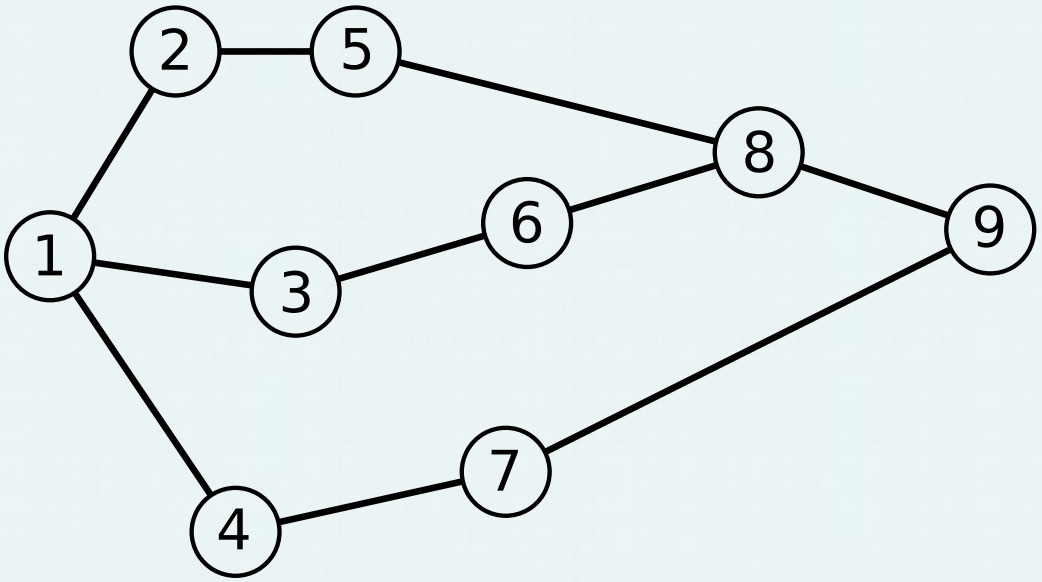
- Flooding
- Pro
  - No discovery needed
  - Always best route found
- Cons:
  - No discovery
  - Horribly inefficient
  - Loops…..

Efficient Flooding

- Need to prevent Loops!
- Solution:
  - Interval counter in packet
  - Every node remembers
    - Original sender ID (e.g. MAC)
    - Interval counter (1, 2, 3, ..)
  - => Destination Sequence Number

Example

Table of Node 1:



Dest.	Next hop	Metric (Hops)	Seq. No.
1	1	0	17
2	2	1	2
3	3	1	15
4	4	1	43
5	2	2	9
6	3	2	58
7	4	2	20
8	3	3	23
9	4	3	13

Discovery By Flooding

- Hello Packets
  - Usually propagated throw the whole network
    - (e.g. batman-adv: OGM, OLSR: HNA)
  - Send in intervals
    - Depends on the expected mobility / on/off rate
- Other discovery mechanisms
  - Distributed Hash Table on mesh networks
  - e.g. for ARP optimization of batman-adv

Implementation for Layer 2

- e.g. batman-adv
- Emulates a virtual network switch
- Easy roaming
- Needs to be implemented as Kernel Module
- Needs to handle lower level protocols (Multicast stuff: ARP, NDP, DHCP)
    - Gets noisy if you miss stuff, otherwise it needs to be flooded

Implementation on Layer 3

- e.g. OLSR, BGP, BMX7
- Easier to implement
- User space makes it more portable to other platforms
- No roaming without hacks

Some Mesh Routing Software

- MANET
  - **Batman-adv**: Kernel module / Distance Vector / Layer 2
  - **Babel**: User Space / Distance Vector / Layer 3
  - **OLSR**: User Space / Link State / Layer 3
  - **BMX7**: User Space / Distance Vector / Layer 3 / Source Routing / focus on Policy Routing for security
- Other
  - **BGP**: organizational tree / Source Routing / Distance Vector / Layer 3 / focus on Policy Routing for peering costs
  - **OSPF**: Spanning tree / Distance Vector / Layer 3

Scalability

- Most MANET protocols that use flooding
  - Management traffic growths exponentially O(n^1..2)
  - ~1000 Nodes is the limit
  - Optimizations (clustering, caching) helps by a factor
- Geographic Routing has potential
  - If it works...

Geographic Routing (1)

- Every Node has coordinate
  - GPS or virtual (better/cheaper)
- Every Node exchanges management traffic with neighbors only
  - Constant management traffic
  - Independent of network size!
- Promises really big mobile adhoc mesh networks

Geographic Routing (2)

- Routing is usually Greedy
- Use local information for routing decision
- Hope to archive good global routing path
- Might get stuck in local minimum
- Backtracking fallback (e.g. Face Routing)

RIP

- **R**outing **I**nformation **P**rotocol
- Precursor of OSPF
- Distance Vector
- hop count metric
  - Max 16 (used to prevent loops!)
- Broadcast whole table every 30 Seconds
  - Slow convergence & inefficient

AODV

- **A**d-hoc **O**n **D**emand Distance **V**ector Routing
  - Often used as citation in academia
  - Descendant of DSDV
- Reactive (“On Demand”)
- Route Request Packet
- Router Reply Packet
- (2003)

DSDV

- Destination Sequence Distance Vector
- Bellman – Ford routing algorithm

The End

Scalability

- MANET protocols max out at ~1000 nodes
- Lower Barrier O(log n)
  - At least need to address every single node
- The capacity of wireless networks (2000 - Guptar, Kumar)
  - Wireless capacity decreases because of transmission overlap / noise
  - Does not apply for when mesh clouds are connected with directional connections

Random Notes

- Clustering is a more flexible form of a spanning tree