15-441: Computer Networking

Lecture 24: Ad-Hoc Wireless Networks

Scenarios and Roadmap

- **Point to point wireless networks (last lecture)**
  - Example: your laptop to CMU wireless
  - Challenges: Poor and variable link quality, hidden and exposed terminals
- **Ad hoc networks (no infrastructure)**
  - Example: military surveillance network
  - Extra challenges: Routing and possible mobility
- **Sensor networks (ad hoc++)**
  - Example: network to monitor temperatures in a volcano
  - Extra challenge: serious resource constraints
- **Vehicular networks (ad hoc+++)**
  - Example: vehicle-2-vehicle game network
  - Extra challenge: <u>extreme</u> mobility

Wireless Challenges (review)

- Interference causes losses, which TCP handles poorly
  - Collisions
  - Multipath interference
  - Environmental (e.g. microwaves)
  - Hidden & exposed terminals
- Contention makes it slow
- Solutions at the Link Layer
  - Local retransmissions
  - RTS/CTS

Ad Hoc Networks

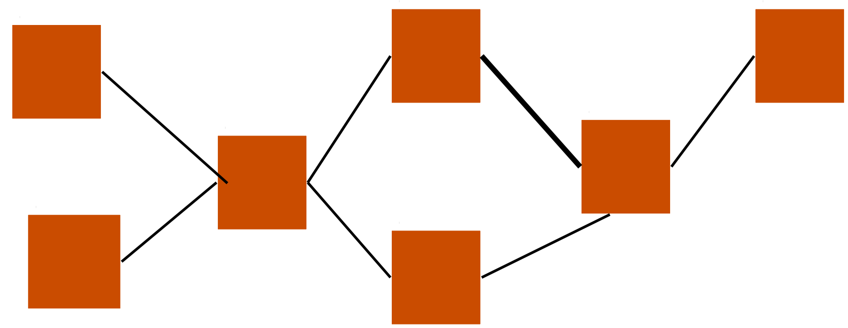
- All the challenges of wireless, plus:
  - No fixed infrastructure
  - Mobility (on short time scales)
  - Chaotically decentralized
  - Multi-hop!
- Nodes are both traffic sources/sinks and forwarders, no specialized routers
- The biggest challenge: <u>routing</u>

Ad Hoc Routing

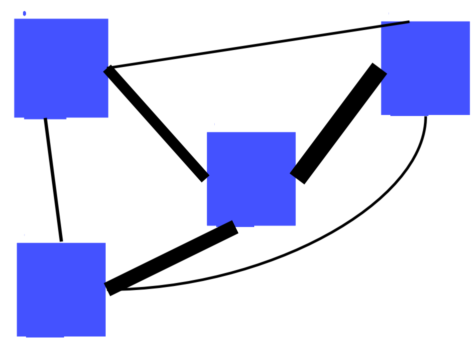
- Find multi-hop paths through network
  - Adapt to new routes and movement / environment changes
  - Deal with interference and power issues
  - Scale well with # of nodes
  - Localize effects of link changes

Traditional Routing vs Ad Hoc

- Traditional network:
  - Well-structured
  - ~O(N) nodes & links
  - All links work ~= well



- Ad Hoc network
  - O(N^2) links - but most are bad!
  - Topology may be really weird
    - Reflections & multipath cause strange interference
  - Change is frequent



Problems Using DV or LS

- DV loops are very expensive
  - Wireless bandwidth << fiber bandwidth…
- LS protocols have high overhead
- N^2 links cause very high cost
- Periodic updates waste power
- Need fast, frequent convergence

Proposed Protocols

- Destination-Sequenced Distance Vector (DSDV)
  - Addresses DV loops
- Ad Hoc On-Demand Distance Vector (AODV)
  - Forwarders store route info
- Dynamic Source Routing (DSR)
  - Route stored in the packet header

- Let's look at DSR

DSR

- Source routing keeps changes local
  - Intermediate nodes can be out of date
- On-demand route discovery
  - Don’t need periodic route advertisements
- (Design point: on-demand may be better or worse depending on traffic patterns…)

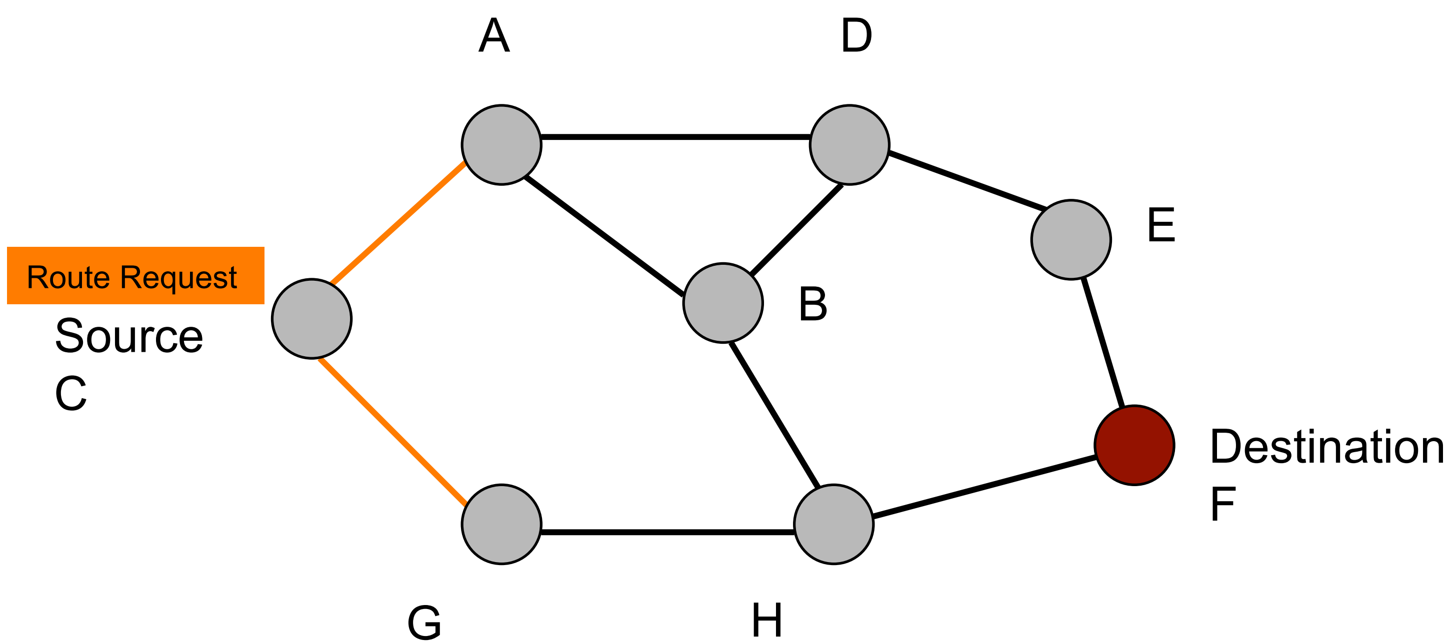
DSR Components

- Route discovery
  - The mechanism by which a sending node obtains a route to destination
- Route maintenance
  - The mechanism by which a sending node detects that the network topology has changed and its route to destination is no longer valid

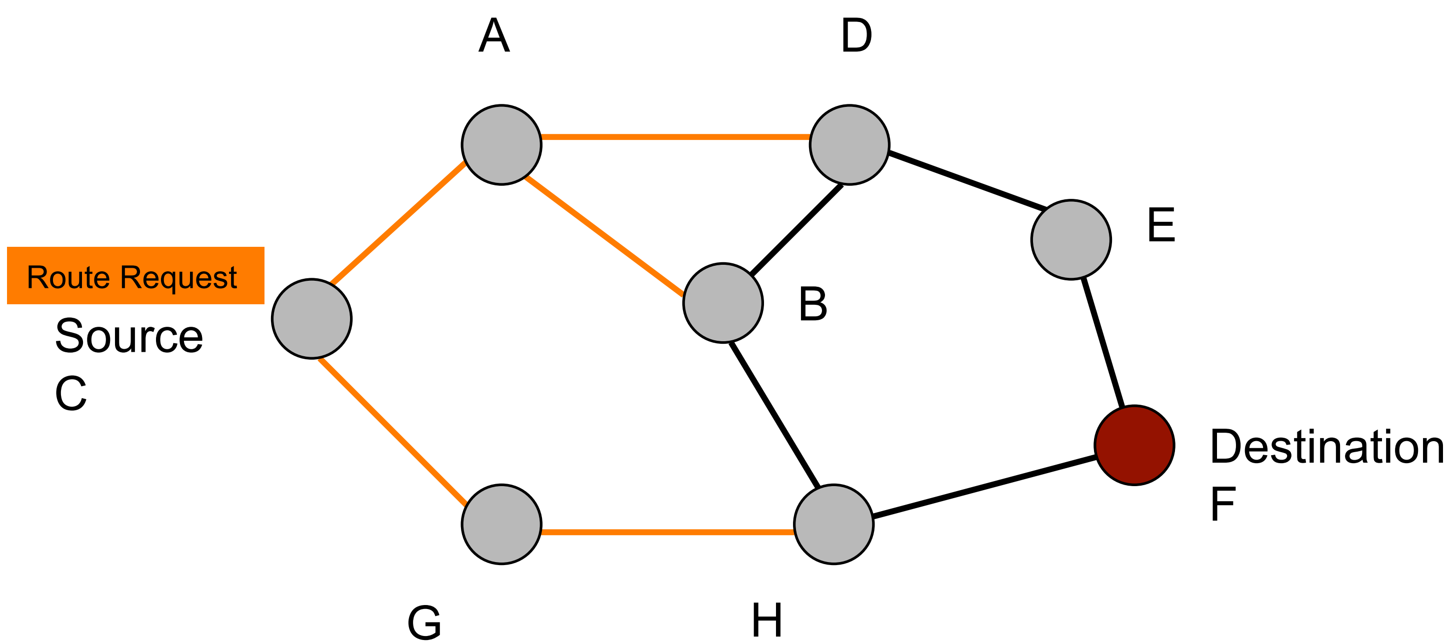
DSR Route Discovery

- Route discovery - basic idea
  - **Source** broadcasts route-request to **Destination**
  - Each node forwards request by adding own address and re-broadcasting
  - Requests propagate outward until:
    - Target is found, or
    - A node that has a route to Destination is found

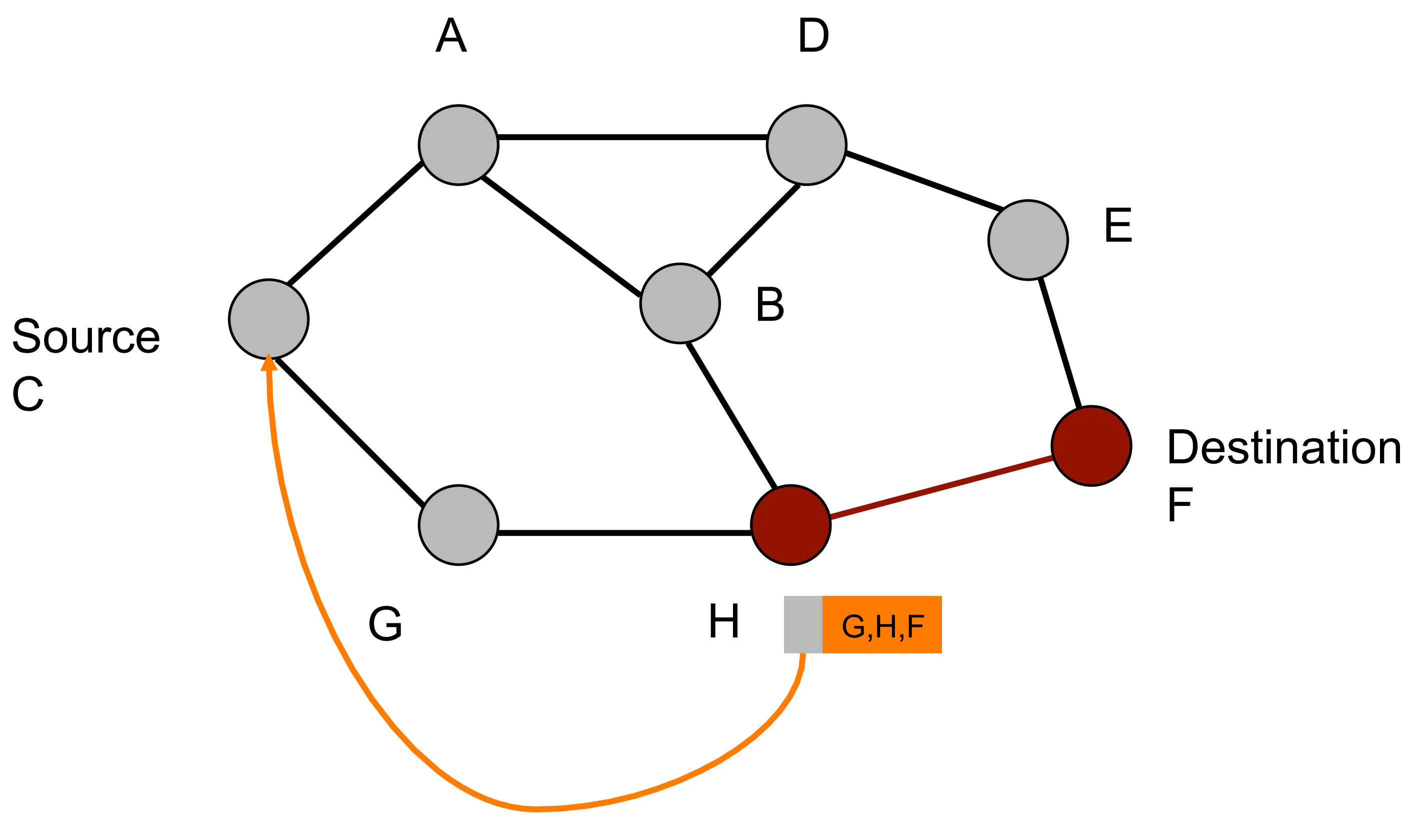
C Broadcasts Route Request to F



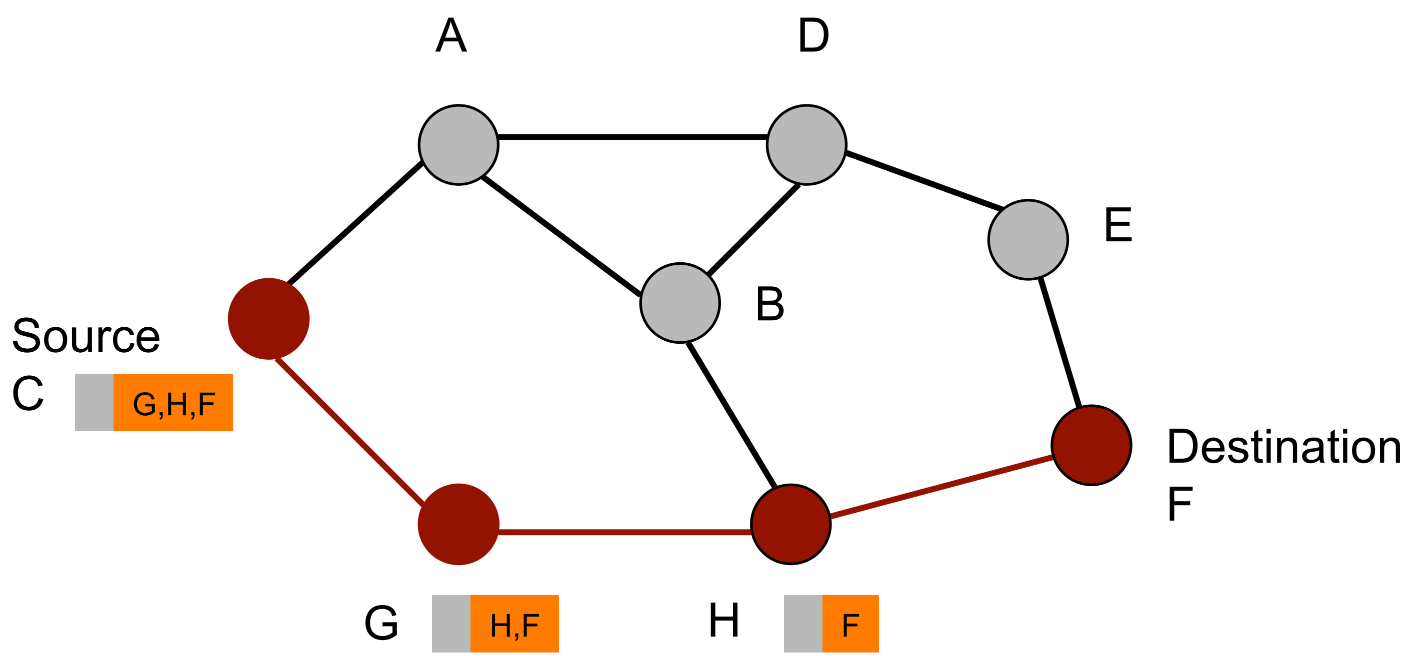
C Broadcasts Route Request to F



H Responds to Route Request



C Transmits a Packet to F



Forwarding Route Requests

- A request is forwarded if:
  - Node doesn't know the destination
  - Node not already listed in recorded source route (loop avoidance)
  - Node has not seen request with same sequence number (duplicate suppression)
  - IP TTL field may be used to limit scope
- Destination copies route into a Route-reply packet and sends it back to **Source**

Route Cache

- All source routes learned by a node are kept in Route Cache
  - Reduces cost of route discovery
- If intermediate node receives RR for destination and has entry for destination in route cache, it responds to RR and does not propagate RR further
- Nodes overhearing RR/RP may insert routes in cache

Sending Data

- Check cache for route to destination
- If route exists then
  - If reachable in one hop
    - Send packet
  - Else insert routing header to destination and send
- If route does not exist, buffer packet and initiate route discovery

Discussion

- Source routing is good for on demand routes instead of a priori distribution
- Route discovery protocol used to obtain routes on demand
  - Caching used to minimize use of discovery
- Periodic messages avoided
- But need to buffer packets
- How do you decide between links?

Forwarding Packets is Expensive

- Throughput of 802.11b =~ 11Mbits/s
  - In reality, you can get about 5.
- What is throughput of a chain?
  - A -> B -> C ?
  - A -> B -> C -> D ?
  - Assume minimum power for radios.
- Routing metric should take this into account

ETX Routing metric

- Measure each link's delivery probability with broadcast probes (& measure reverse)
- P(delivery) = 1 / (df * dr)   (ACK must be delivered too)
- Link ETX = 1 / P(delivery)
- Route ETX = sum of link ETX
- (Assumes all hops interfere - not true, but seems to work okay so far)

Capacity of Multi-Hop Network

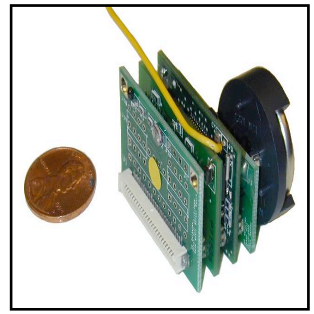
- Assume N nodes, each wants to talk to everyone else. What total throughput (ignore previous slide to simplify things)
  - O(n) concurrent transmissions. Great! But:
  - Each has length O(sqrt(n)) (network diameter)
  - So each Tx uses up sqrt(n) of the O(n) capacity.
  - Per-node capacity scales as 1/sqrt(n)
    - Yes - it goes down! More time spent Tx'ing other peoples packets…
- But: If communication is local, can do much better, and use cool tricks to optimize
  - Like multicast, or multicast in reverse (data fusion)
  - Hey, that sounds like … a sensor network!

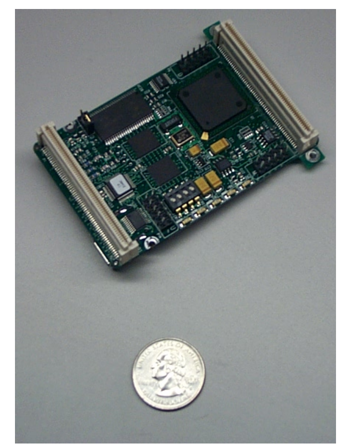
Sensor Networks – Smart Devices

- First introduced in late 90’s by groups at UCB/ UCLA/USC
- Small, resource limited devices
  - CPU, disk, power, bandwidth, etc.
- Simple scalar sensors – temperature, motion
- Single domain of deployment
  - farm, battlefield, bridge, rain forest
- for a targeted task
  - find the tanks, count the birds, monitor the bridge
- Ad-hoc wireless network

Sensor Example – Smart-Dust

- Hardware
  - UCB motes
  - 4 MHz CPU
  - 4 kB data RAM
  - 128 kB code
  - 50 kb/sec 917 Mhz radio
  - Sensors: light, temp.,
    - Sound, etc.,
  - And a battery.





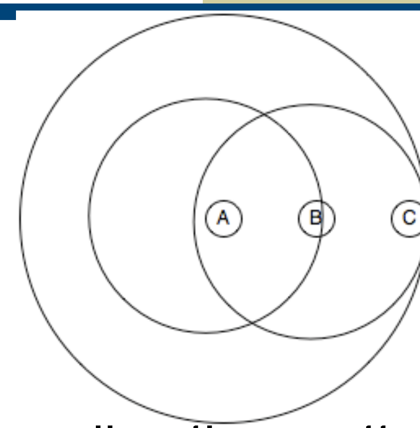
Sensors, Power and Radios

- Limited battery life drives most goals
- Radio is most energy-expensive part.
- 800 instructions per bit. 200,000 instructions per packet. (!)
- That's about one message per second for ~2 months if no CPU.
- Listening is expensive too. :(

Sensor Nets Goals

- Replace communication with computation
- Turn off radio receiver as often as possible
- Keep little state (limited memory).

Power



- Which uses less power?
  - Direct sensor -> base station Tx
    - Total Tx power: distance^2
  - Sensor -> sensor -> sensor -> base station?
    - Total Tx power: n * (distance/n) ^2 =~ d^2 / n
  - Why? Radios are omnidirectional, but only one direction matters. Multi-hop approximates directionality.
- Power savings often makes up for multi-hop capacity
  - These devices are *very* power constrained!
- Reality: Many systems don’t use adaptive power control. This is active research, and fun stuff.

Example: Aggregation

- Find average temperature in GHC 8th floor.
  - Naïve: Flood query, let a collection point compute avg.
    - Huge overload near the CP. Lots of loss, and local nodes use lots of energy!
- Better:
  - Take local avg. first, & forward that.
    - Send average temp + # of samples
  - Aggregation is the key to scaling these nets.
- The challenge: How to aggregate.
  - How long to wait?
  - How to aggregate complex queries?
  - How to program?

Beyond Sensors – Vehicular Ad-Hoc Networks

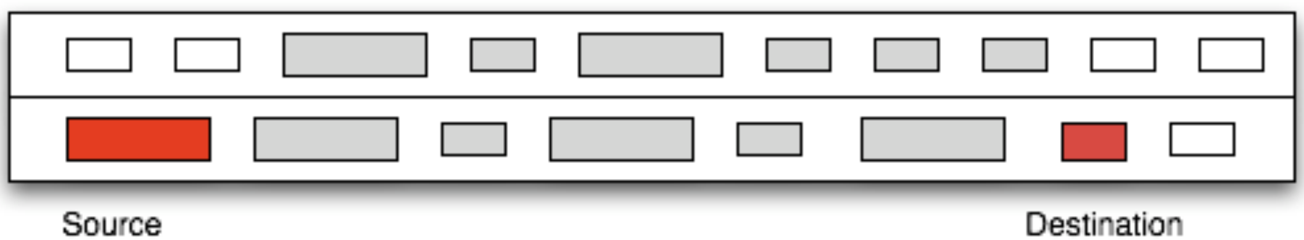
- Aggregation is not everything
- Power and computation constraints limiting
- What can we use as highly mobile and powerful ad hoc network nodes? <u>Cars!</u>
- Potential applications for VANETs
  - Collision avoidance
  - Virtual traffic signals
  - (Semi-)Autonomous driving
  - Infotainment

Vehicular Networks – Challenges?

- Extreme mobility
  - DSR won’t work if the routes keep changing
- Scale
  - Possibly the largest ever ad-hoc networks
- Topology
  - Deployment/density not controlled by designer (e.g., highway vs city)
  - Gradual deployment (new cars equipped from the factory in the near future)

VANET Routing – Simple case

- Topology based routing
  - DSR won't work because the nodes keep changing
  - Can form clusters and route through cluster heads (LORA_CBF)
- Geographical routing
  - Use relative position between node, source and destination to, on the fly, decide whether to forward or not (GPSR)



VANET Routing – General case

- Cities, rural areas
  - Topology-based routing fails, geographical routing harder
    - Local minima/network holes: no neighbor is closer to the destination than we are
    - Greedy Perimeter Stateless Routing (GPSR) routes around the perimeter
  - What we would really want
    - To have a density map of the network to help us choose forwarders

VANET Routing – General case

- Learning about node density in VANETs
  - Use road maps and statistical traffic information (A-CAR)
    - Coarse-grained
  - Local, neighbor based estimation
    - Local optimum != global optimum
  - Online, large scale estimation
    - High overhead
- No perfect solution – open research topic

Important Lessons

- Wireless is challenging
  - Assumptions made for the wired world don’t hold
- Ad-hoc wireless networks
  - Need routing protocol but mobility and limited capacity are problems
  - On demand can reduce load; broadcast reduces overhead
- Special case 1 – Sensor networks
  - Power is key concern
  - Trade communication for computation
- Special case 2 – Vehicular networks
  - No power constraints but high mobility makes routing even harder, geographical routing