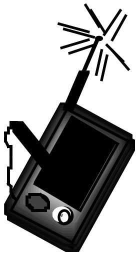
CS 294-7: Routing in Packet Radio Networks

Professor Randy H. Katz

CS Division

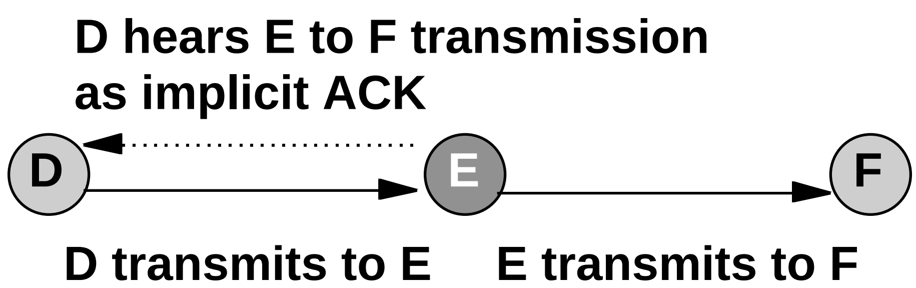
University of California, Berkeley

Berkeley, CA 94720-1776

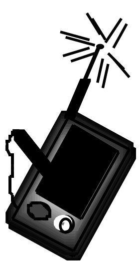


Packet Radio Network Schemes (ARPA PR Program)

- **Passive Acknowledgments**
  - **Half duplex operation: transmit packet, go into receive mode, receive ack, receive next packet, go into transmit mode, repeat**
  - **Original sender hears forwarding transmission from next hop node in the route:**

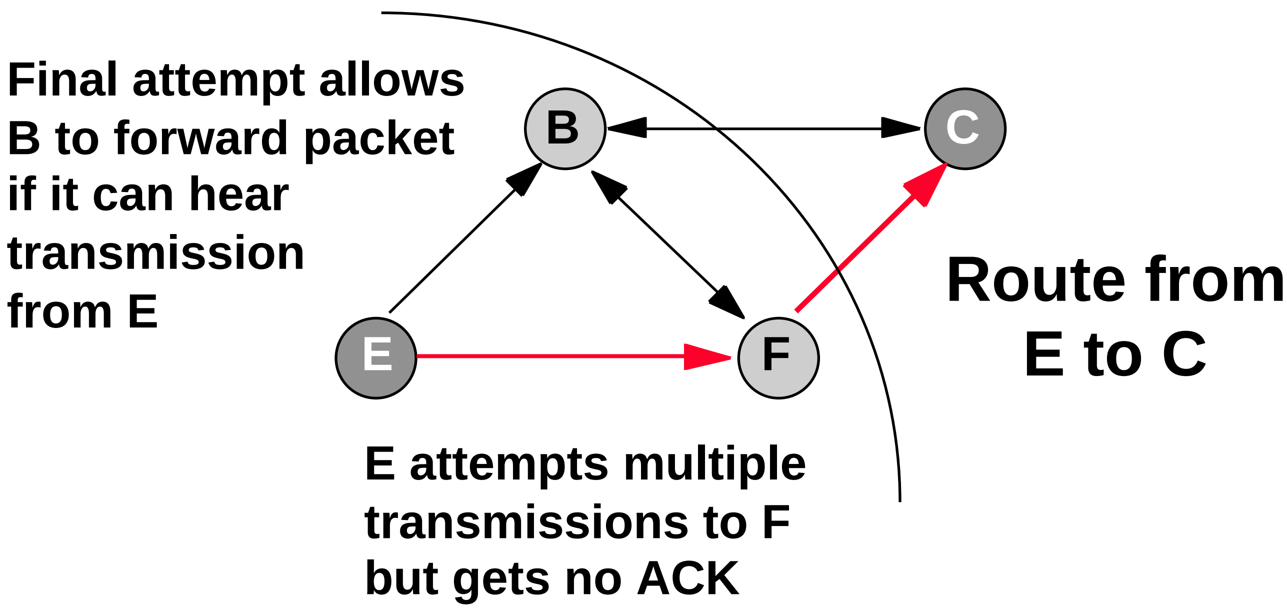


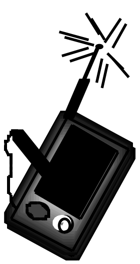
  - **Power control: transmit with enough power to be heard at D as well as F**



Packet Radio Network Schemes (ARPA PR Program)

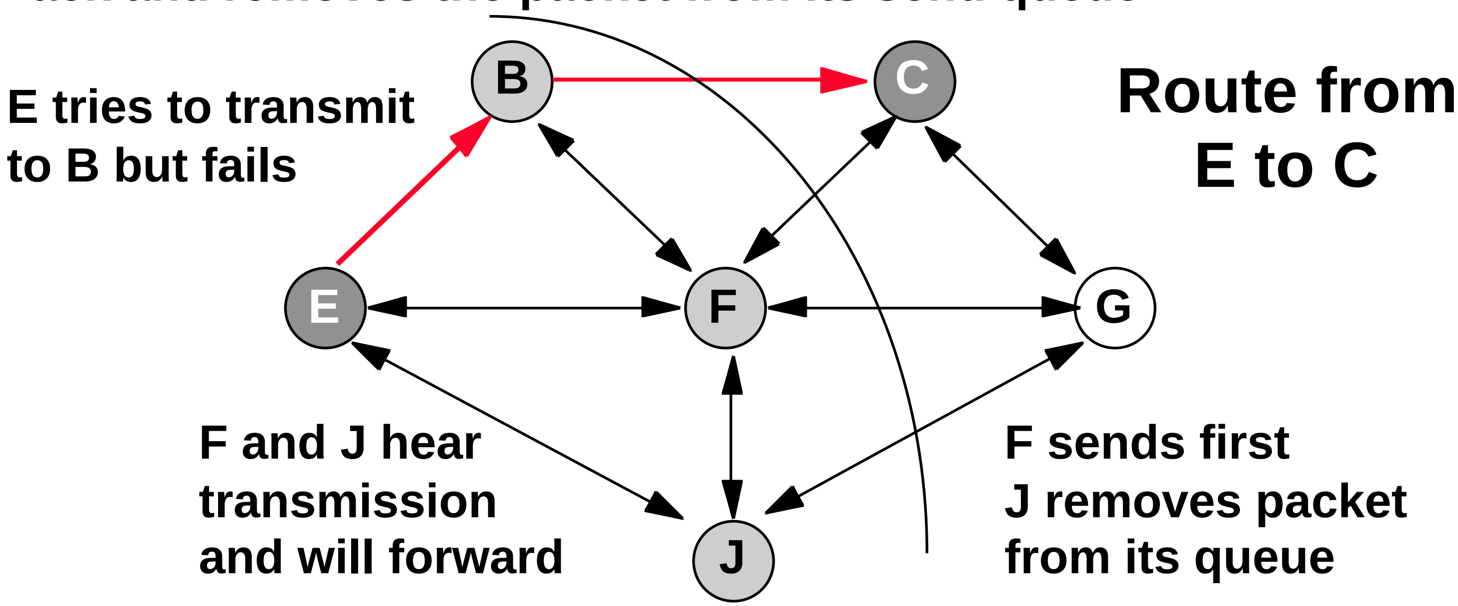
- **Alternative Routing on Retransmission**
  - Packet contains header for next node on route
  - After multiple tries, if no ack, header changed to indicate that ANY closer node to ultimate destination may forward the packet

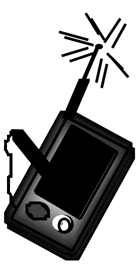




Packet Radio Network Schemes (ARPA PR Program)

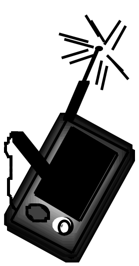
- **Filtering Based on Overheard Traffic**
  - Alternative routing scheme can cause undesirable packet flooding (i.e., multiple nodes forwarding on the same packet)
  - If packet is queued at a node X, and X hears the same packet sent from a different node, it assumes an implicit ack and removes the packet from its send queue





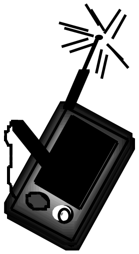
Packet Radio Network Issues

- **Channel Access and Hidden Terminals**
- **Throughput**
  - Assume uniform traffic, slotted Aloha access method, K average neighbors, and H average # of hops
  - For single hop, implies .36/K * available ch b/w
  - For multihop, implies .36/(H * K)
  - Must also add ack and retransmission overheads
  - For a given radio on a packet path, must receive packet, retransmit packet, and receive ack: max 1/3 of available b/w
  - Clearly works best when traffic is bursty!
- **Connectivity**
  - Poor channels implies link layer ack schemes
  - Exchange of connectivity packets:
    Link quality = *f*(packets received, total packets sent)
  - Must be able to deal with network partitions



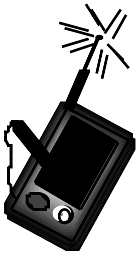
Packet Radio Network Issues

- **Broadcast Nature of Radio Channels**
  - **Nodes can overhear the forwarding of packets by other nodes**
  - **This capability can be exploited to**
    - » **Send network-control traffic to all neighbors**
    - » **Support algorithms for broadcasting user traffic**
    - » **Circumvent link failure by finding an alternative node to route the packet**



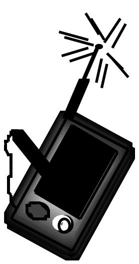
DARPA Packet Radio Network

- Low cost packet radios in 1983 technology
  - DS spread spectrum
  - Multiple transmission rates (100, 400 kbps)—trade error coding for b/w when the link is good
  - Adaptable FEC coding, can be changed on a packet by packet basis
  - Up to 10 km range
  - Broadcast and receiver-directed packet transmission
    - » Former uses a code sequence known to all radios
    - » Latter uses a unique code for the target receiver; Other receivers won't be able to extract bit sync, and will be free to receive another packet



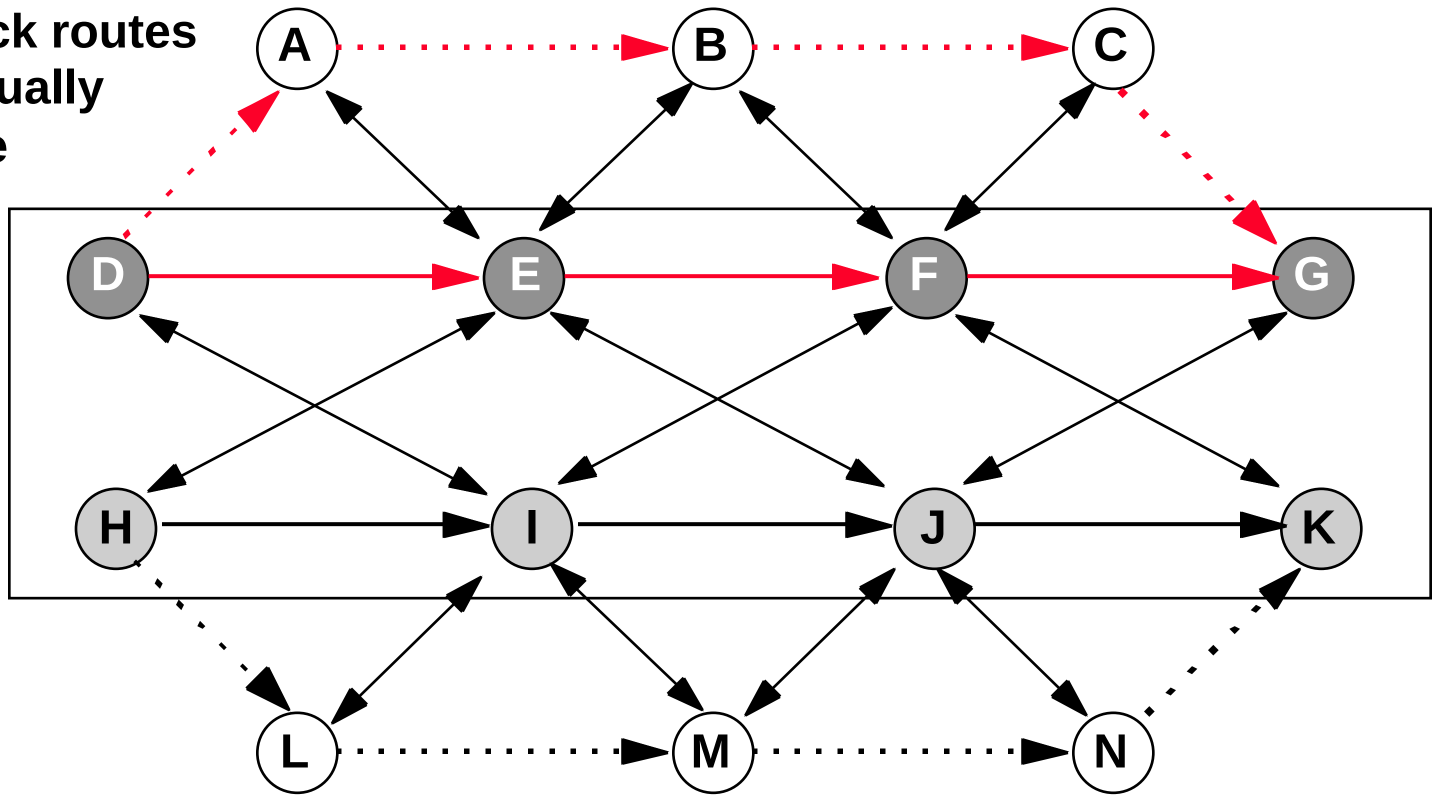
Routing in Small to Medium Sized Networks

- **Optimal Routing Schemes:**
  - Need estimates of network flows, fixed topology, residual capacity of links or incremental delay of links
  - Assumes independence from traffic on other links/paths
- **Difficult to use because:**
  - Info about network flows NOT generally available in datagram networks
  - Underlying network topology can change rapidly
  - Delay/capacity parameters change more rapidly than topology, yielding increased overhead to keep information up to date
  - Information for routing purposes likely to be out of date by the time it is fully disseminated throughout the network
  - Delay/capacity of a link is a function of traffic on other links

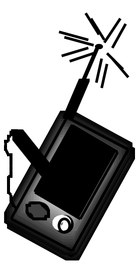


Routing in Small to Medium Sized Networks

Transmissions along the red and black routes will mutually interfere

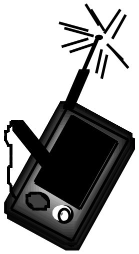


Alternative routes with more hops but less interference



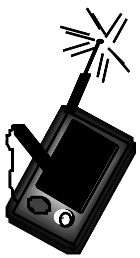
Alternative Routing Metrics

- Least Interference Routing (LIR)
  - Determine routing cost based on the number of radios that can overhear transmissions on the link
  - Only needs nearest neighbor information
- Max-Min Residual Capacity Routing (MMRCR)
  - Compute routing cost based on traffic dependent metric that is a function of probability of successful transmission and interference
- Least Resistance Routing (LRR)
  - Routing cost is a function of interference, accounting for both other radio transmission and jamming



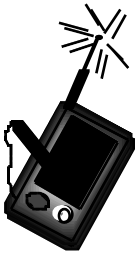
Link Connectivity and Route Calculation

- **Network information stored in**
  - Neighbor table
  - Tier table
  - Device table
- **Neighbor table**
  - Broadcast a Packet Radio Organization Packet (PROP) every 7.5 seconds
    - » Neighbors that hear a PROP make entry in their neighbor tables
    - » When nodes hears a PROP, it updates its neighbor table
    - » Transmitted data packets also used to build neighbor table
  - Also tracks bidirectional quality of links with neighbors (retransmission counts)



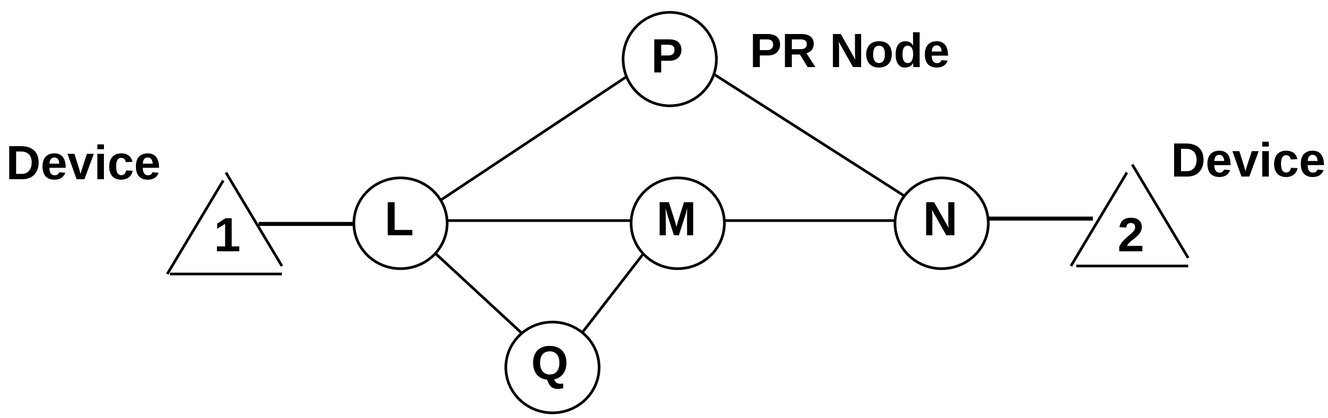
Link Connectivity and Route Calculation

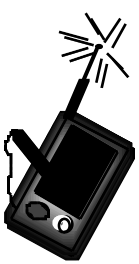
- **Tier Table**
  - Every packet radio knows the “best” next node on the route from it to a given destination node
  - Tier 1 = 1 hop neighbors
  - These neighbors send out their PROPs indicating that they are one hop from the originator
    - » At next step, receivers of these PROPs know that they are 2 hops away from the originator
    - » Process continues until every radio its distance in tiers from every other radio
  - “Best”: shortest route with “good” connectivity
    - » To change table, must discover a new node with better link quality and lower tier number than currently recorded next node
  - Also disseminate information about bad links in PROP messages



Link Connectivity and Route Calculation

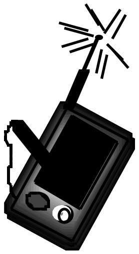
- **Device Table**
  - Logical addressing: maps device to a packet radio
  - Information about the radio’s attached device is included in PROP messages
  - This allows new radios to be attached to devices and vice versa
  - Such correspondences are maintained in the device table at each packet radio





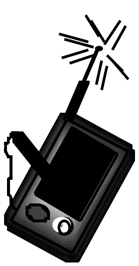
Forwarding Protocols

- Unlike PROPs, user packets are *not* flooded in conserve available bandwidth
- Packet Headers:
  - End-to-end header
    - » <Src Device ID, Dest Device ID, Type of Service Flag>
    - » ToS: indicate low latency/low reliability, e.g., speech
  - Routing header
    - » Src PR ID, Seq No, Speech ToS flag, Prev PR ID (for acks), Prev PR transmission count, Transmitting PR ID, Transmitting PR transmit count (for pacing), Next PR ID, Lateral alternative routing flag, Alternative routing request flag, Tier, Dest PR ID



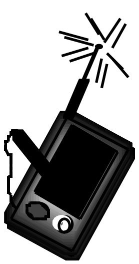
Forwarding Protocol

- **Device 1 --> Device 2 via PRs L, M, N**
- **Device 1 --> PR L**
  - **Device sends packet PR L via its wired connection; Prepare packet to forward on to PR N via PR M:**
    - » **Dest PR ID <- N**
    - » **Prev PR ID <- null**
    - » **Trans PR ID <- L**
    - » **Next PR ID <- M (known from tier table)**
    - » **Tier <- 2 (from tier table)**



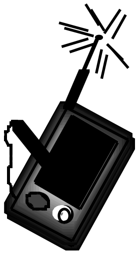
Forwarding Protocol

- **PR L --> PR M**
  - **PR M receives packet over the air**
  - **Next PR ID = M, this PR should process the packet**
  - **Prepare to forward packet on to PR N:**
    - » **Prev PR ID <- L**
    - » **Transmitting PR ID <- M**
    - » **Next PR ID <- N (known from tier table)**
    - » **Tier <- 1 (from tier table)**
  - **Transmit packet to PR N … and any other PR within range, including L! This is an example of the passive acknowledgement.**



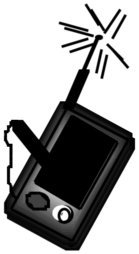
Forwarding Protocol

- **PR M --> PR N**
  - **N receives packet, determines it should process it based on Next PR ID**
  - **Determines that packet should be delivered to the attached Device 2 (from ETE header and device table)**
  - **Wire-line transmits the packet to Device 2**
  - **Sets in header, for the ack message:**
    - » **Prev PR ID <- M**
    - » **Trans PR ID <- N**
    - » **Next PR ID <- null**
    - » **Tier <- null**
    - » **Ack message is "short", consisting only of header**
  - **Note that end PR can't use passive acknowledgement, so is forced to transmit ack message to PR M**



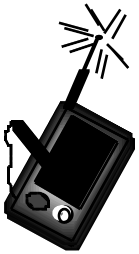
Forwarding Protocol

- **Criteria for recognizing an Ack**
  - **Source PR ID and Seq No match the original packet**
  - **AND must have arrived from further downstream:**
    - » **Transmitting PR ID in ack packet is same as next PR ID in original packet**
    - » **Previous PR ID is same as receiving PR's ID--the forward packet came from this packet radio**
    - » **Ack packet contains a smaller tier number, indicating it got closer to the destination PR**



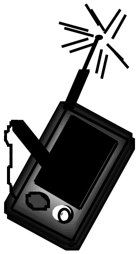
Forwarding Protocol

- **Retransmissions**
  - If a packet is forwarded, and no ack is received, the packet will be retransmitted after a time out
  - Will do this six times before giving up
  - Interval between retransmission based on pacing protocol, and grows with each successive unsuccessful retransmission
  - At some point, sending PR assumes that it can no longer reach the next radio on the hop and sets its connectivity to that radio to 0
  - FEC and CRC used to reduce the chance of retransmissions



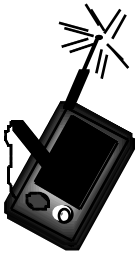
Forwarding Protocol

- **Alternative Routing**
  - **3 unsuccessful retransmissions: request forwarding help**
  - **Alternative routing request flag**
    - » **Receiving PR whose ID ≠ Next PR ID**
    - » **This PR will forward if it's tier table indicates it is less than or equal to the tier in the header (it as close or closer to the destination than the sender)**
    - » **Lateral Alternative Routing Flag is then set to insure that the packet does not loop around at the same tier level for ever--the next PR in the route MUST be closer to the destination**
  - **Duplicate packets filtered out by checking Source PR ID and sequence number field (i.e., Unique Packet ID--UPI)**
    - » **When detected at receiver, will actively ACK sender to squelch retransmissions of duplicate packet**



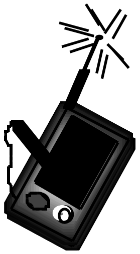
Transmission Protocols

- **Pacing protocol**
  - **Flow and congestion control mechanisms**
  - **Also promotes fair use of the radio spectrum**
    - » **Single Threading**
    - » **Forwarding Delay Measurement**
    - » **Measurement of Retransmissions**



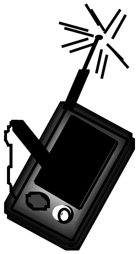
Transmission Protocols

- **Single Threading**
  - Last packet sent to PR must be ack'd before next packet is sent to the same PR
    - » Passive acks imply that next hop PR now ready to accept a new packet
    - » Deflects congestion bottleneck away from source PR
- **Forwarding Delay**
  - Affects the setting of retransmission intervals
  - Ack TS – Original Next Hop Transmission TS
  - Includes processing, queing, carrier sense/random access, transmission delay from neighboring PR
  - Exponentially smoothed to get short term history of delay likely to be valid in next transmit/ack cycle



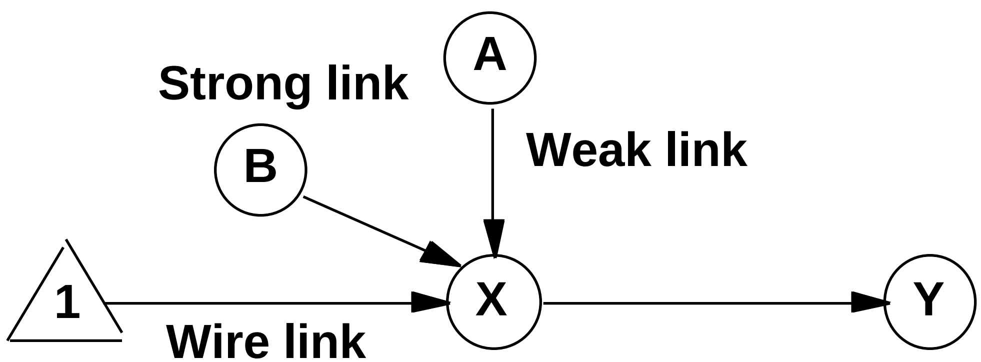
Transmission Protocols

- **Pacing Function**
  - L transmits to M
  - M transmits to N, passively acks to L
  - M receives ack from N
  - M now ready to receive a new packet from L
  - Effectively, no PR can transmit more than 1/3 of the time!
  - Multiply measured delays by 3 to reflect this separation of transmissions among neighbor PRs



Transmission Protocols

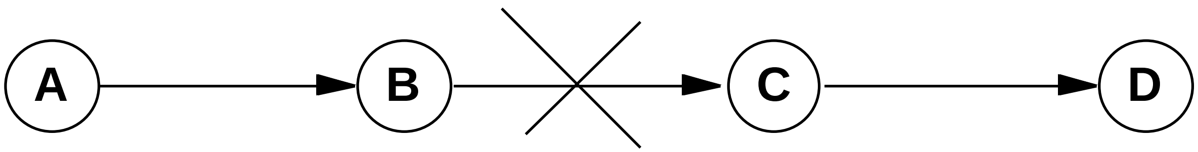
- **Fairness Queuing**
  - **Packets handled FIFO, except:**
    - » **Packets for fast links given higher priority than packets for slow links (speed defined in a pacing sense)**
    - » **Packets from slow links inserted before second packet from a fast link**



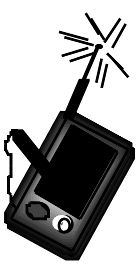
Packets from A inserted before second queued packet from either B or 1

Loop Formation and How to Avoid It

- Node must report bad route info via PROP *before* accepting a new route

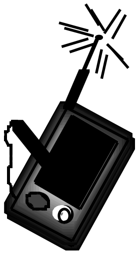


  - B to D: 2 hops, A to D: 3 hops
  - B asks for new route to D from A: thinks it has a 4 hop route to D!
  - When A receives PROP from B, it thinks it has a 5 hop route to D!
  - This routing loop might persist, with the hops incrementing until they reach some maximum value



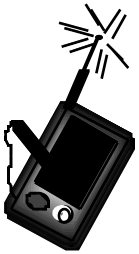
Extensions

- **Propagating Information about Bad Routes**
  - Original algorithm: bad route information must be reported to neighbors before good route accepted from neighbor
  - Problem: information about bad routes propagated in single PROP messages; what if this gets lost?
  - Solutions:
    - » Increase # of PROP messages, but that uses bandwidth
    - » Disseminate the information multiple (3) times, but place it in packet headers rather than in independent PROPs



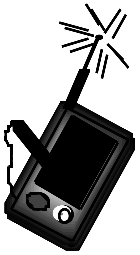
Extensions

- **Multiclass Routing**
  - **Original algorithm: all links in route have *confirmed* bidirectional connectivity**
  - **Multiclass routing allows *unconfirmed* routes**
    - » **First class: confirmed**
    - » **Second class: unconfirmed (typically these are links "coming up" in the network)**
    - » **"Marked for erasure": routes with one or more links being deleted from the network**
  - **First class routes of any length preferred; use the one with the shortest hop count**
  - **Report change in class three times before route status can change again**
  - **Marked for erasure routes must be reported three times before actual deletion**

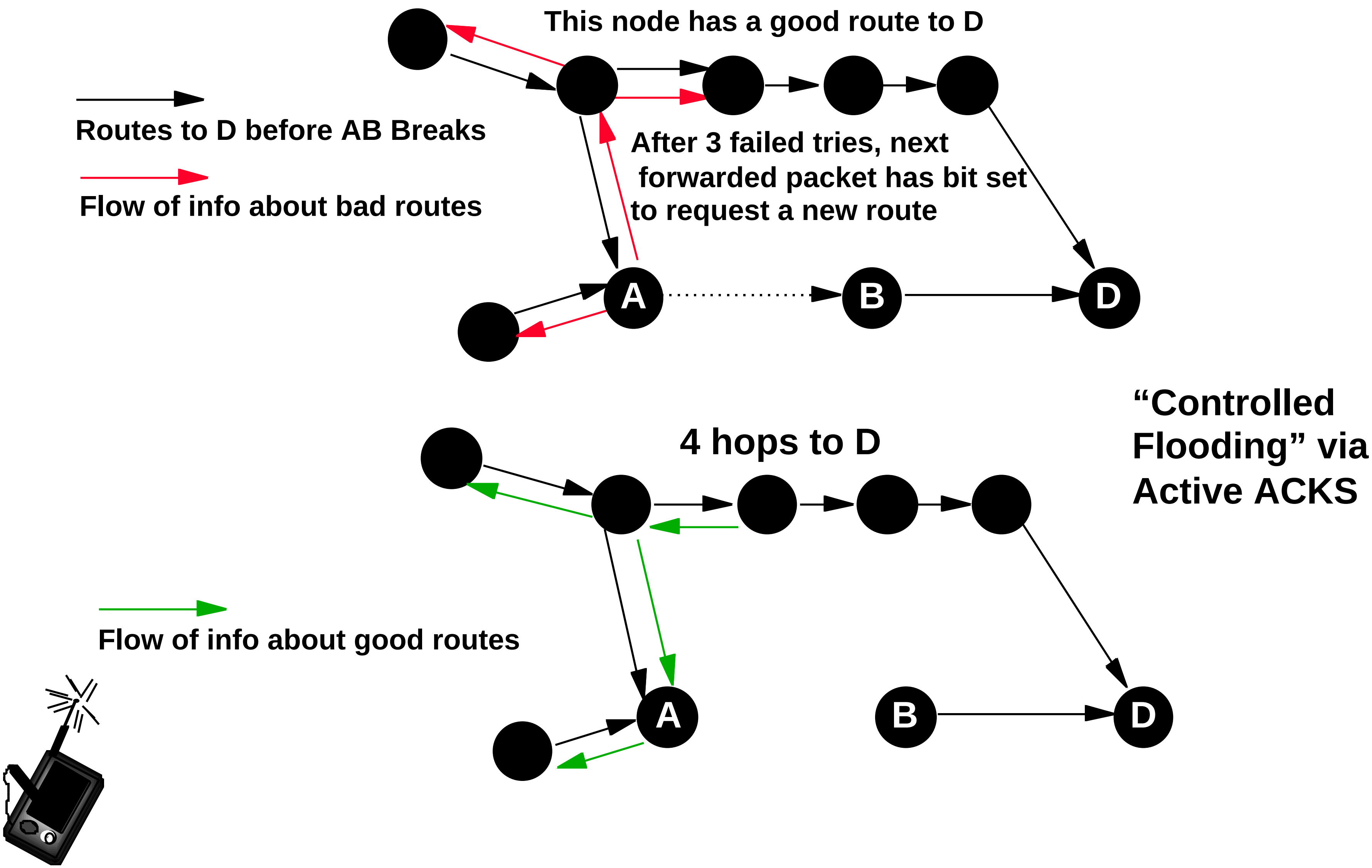


Extensions

- **Event-Driven Updates**
  - **Problem: Slow propagation of lost link information yields route loops and lost traffic**
  - **Basic algorithm:**
    - » **PR X loses route to destination PR Y**
    - » **Receives packet for an unreachable node: active ACKs sender that route has failed**
    - » **Overhearing PRs with X as next hop destination for Y marks the route to be erased**
  - **Extended algorithm:**
    - » **Receiver of active ACK must broadcast the active ACK and mark route to be erased**
    - » **PR that has reported bad route 3 times requests new route from neighbors**
    - » **PR with good route responds via Active ACK to new routing requests**



Event Driven Routing Example



Interactions and Complexities

- Tier and Overheard Routing
  - Packets are transmitted with original route info
  - But this info may change as the topology changes
  - Must update routing info in header before forwarding
- Routing and Duplicate Filtering
  - Must be careful not to filter redirected packets due to routing changes: <Src ID, Seq #, NEXT HOP> determine duplicates
- Partitioned Networks
  - What if no routes to destination because of partitioned network? Limit the number of times a failed route can be advertised to avoid congesting the network