The Basics of Wireless Communication

Octav Chipara

Agenda

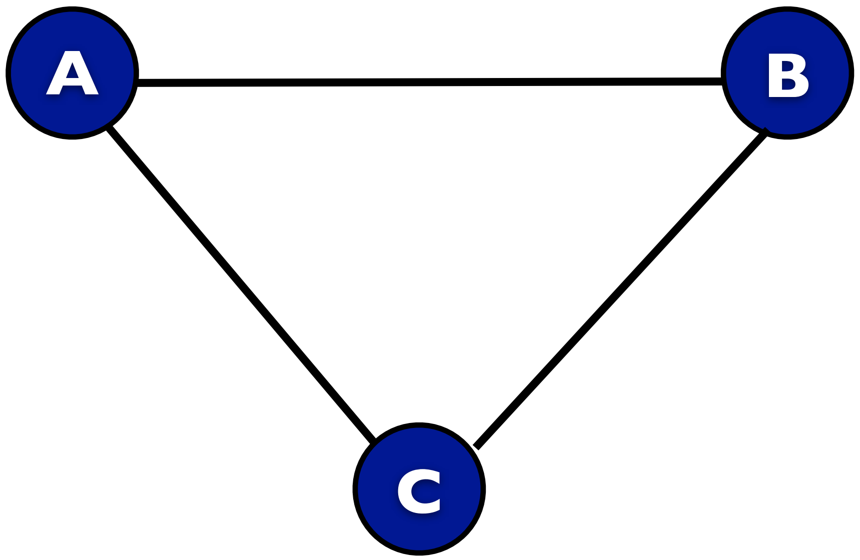
- **Channel model: the protocol model**
- **High-level media access**
  - TDMA, CSMA
  - hidden/exposed terminal problems
- **WLAN**
- **Fundamentals of routing**
  - proactive
  - on-demand

Channel models

- **Channel models - document assumptions of wireless properties**
  - the basis upon which we build and analyze network protocols
- **A good model is one that is**
  - simple - reason effectively about the properties of protocols
  - accurate - capture prevalent properties of wireless channels
  - these requirements are often conflicting
- **Must provide insight into fundamental problems**
  - media access
  - routing
  - congestion
- **Today, simple channel model..., next lecture more realistic models**

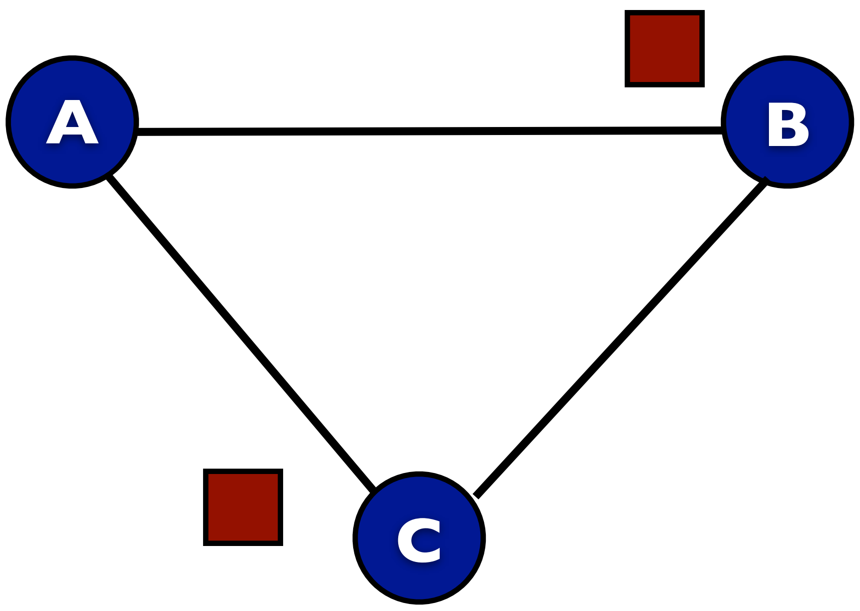
Protocol model

- **Network is modeled as a graph**
  - vertices - all nodes in a graph
  - edges - connect nodes that may communicate
- **Properties:**
  - captures connectivity information
  - packet collisions (**collisions happen only at the receiver**)
  - radios are half-duplex



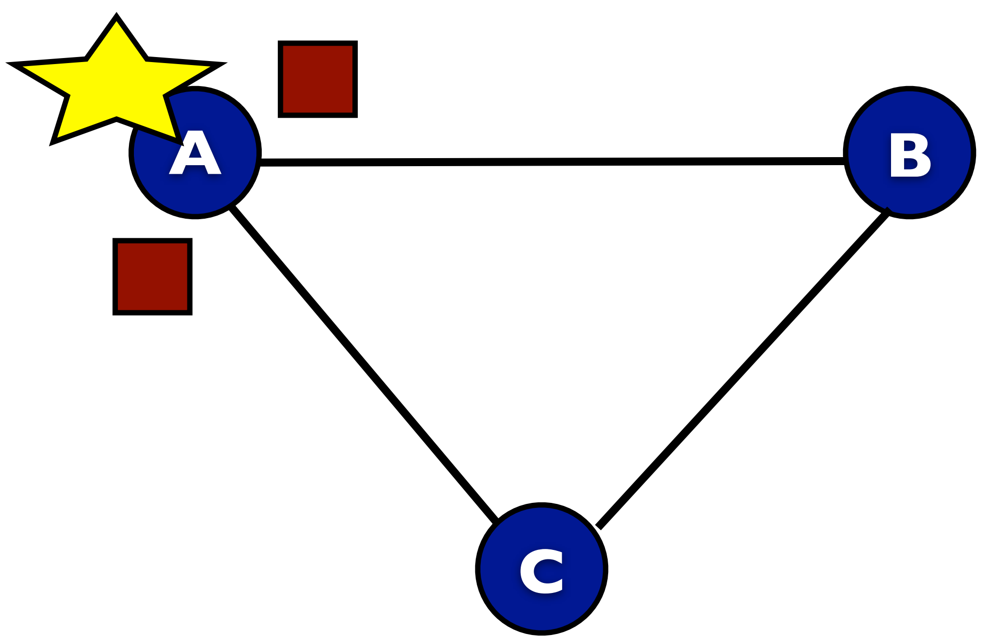
Protocol model

- **Network is modeled as a graph**
  - vertices - all nodes in a graph
  - edges - connect nodes that may communicate
- **Properties:**
  - captures connectivity information
  - packet collisions (**collisions happen only at the receiver**)
  - radios are half-duplex



Protocol model

- **Network is modeled as a graph**
  - vertices - all nodes in a graph
  - edges - connect nodes that may communicate
- **Properties:**
  - captures connectivity information
  - packet collisions (**collisions happen only at the receiver**)
  - radios are half-duplex



Media Access and Control (MAC)

- **Problem: multiple nodes want to transmit concurrently**
  - nodes transmitting concurrently ➔ packet collisions

- **Metrics for characterizing MAC performance**
  - throughput - number of packets delivered per second
  - latency - time to deliver a packet
  - energy efficiency - energy consumed for tx and rx
  - fairness - each node gets its “fair” share of the channel
  - flexibility - how does the MAC handle changes in workload

- **Approaches**
  - CSMA - Carrier Sense Multiple Access
  - TDMA - Time Division Multiple Access

CSMA

- **CSMA - Carrier Sense Multiple Access**
- **Approach:**
    - **1:** node will attempt to transmit after a random delay t ∈ CW
    - **2:** check if channel is available
        - free ➔ perform packet transmission
        - busy ➔ CW = CW * 2, go to step 1
- **Notes:**
    - nodes operate independently!
    - the underlying performance is highly dependent on selecting CW
        - CW - reflects the expected number of contenders for the channel
        - CW increases exponentially [the rate depends on protocol]
    - assumption: the sender can accurately check if channel is free/busy
        - usually holds because:
          receiver sensibility << channel quality required for communication

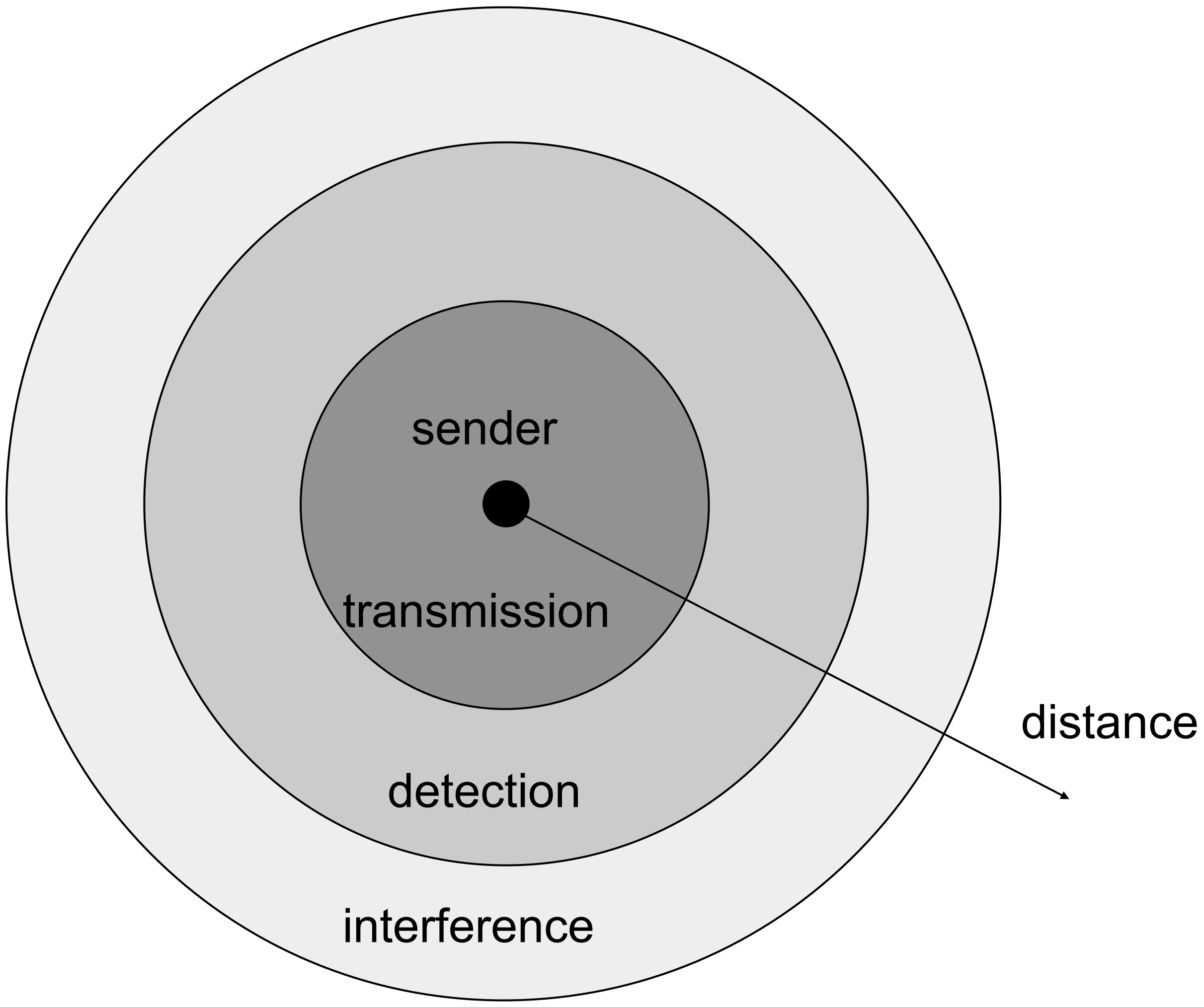
CSMA

- **CSMA - Carrier Sense Multiple Access**
- **Approach:**
  - **1:** node will attempt to transmit after a random delay $t \in CW$
  - **2:** check if channel is available
    - free ➔ perform packet transmission
    - busy ➔ CW = CW * 2, go to step 1

- **Notes:**
  - nodes operate independently!
  - the underlying performance is highly dependent on selecting CW
    - CW - reflects the expected number of contenders for the channel
    - CW increases exponentially [the rate depends on protocol]
  - assumption: the sender can accurately check if channel is free/busy
    - usually holds because:
      receiver sensibility << channel quality required for communication

Signal propagation ranges

- **Transmission range**
  - communication possible
  - low error rate
- **Detection range**
  - detection of the signal possible
  - no communication possible
- **Interference range**
  - signal may not be detected
  - signal adds to the background noise



TDMA

- **TDMA - Time Division Multiple Access**
- **Approach:**
  - **1:** construct a frame in which each node gets a slot to transmit
    - **F** - frame size, **fn** - slot in which node n is assigned to transmit
  - **2:** a node n will transmit at time (**t** mod **F**) = **fn**

- **Notes:**
  - time synchronization is required
  - frame construction requires a global agreement among nodes
  - underlying performance depends on matching a node's workload demand with its slot allocations
    - hard to do due to dynamic workloads and channel properties
  - assumption: only one successful transmission per slot

TDMA

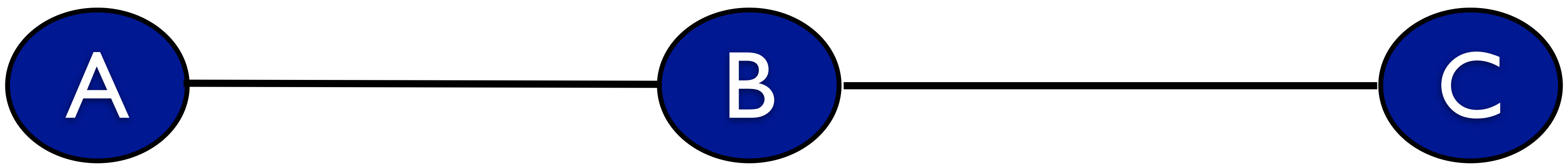
- **TDMA - Time Division Multiple Access**
- **Approach:**
  - **1:** construct a frame in which each node gets a slot to transmit
    - **F** - frame size, **fn** - slot in which node n is assigned to transmit
  - **2:** a node n will transmit at time (**t** mod **F**) = **fn**

- **Notes:**
  - time synchronization is required
  - frame construction requires a global agreement among nodes
  - underlying performance depends on matching a node's workload demand with its slot allocations
    - hard to do due to dynamic workloads and channel properties
  - assumption: only one successful transmission per slot

Single-hop vs. multiple hops

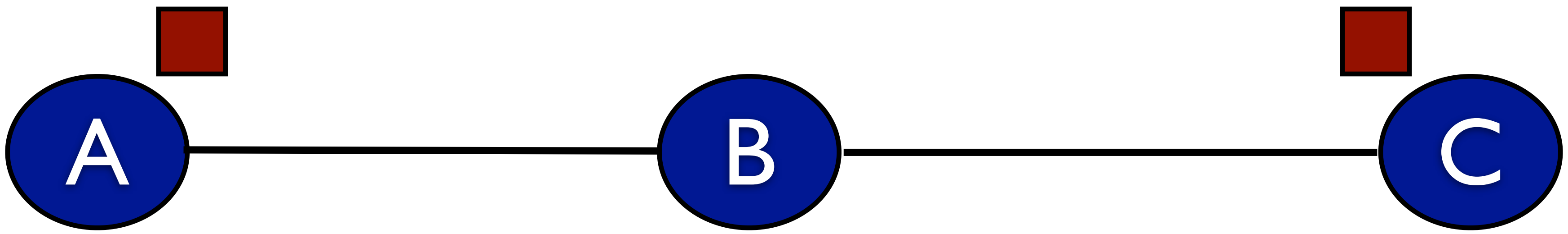
- **Single-hop networks**
  - both CSMA and TDMA protocols are easy to implement
- **Multi-hop networks**
  - important challenges arise due to asymmetrical views of the networks
  - hidden-terminal problem
  - exposed-terminal problem

Hidden terminal problem



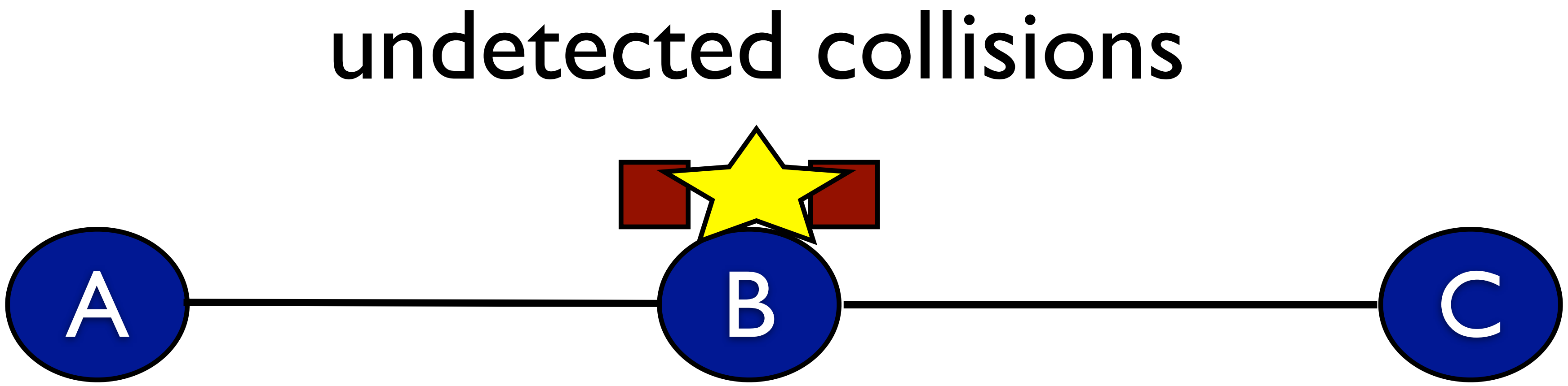
- **Node A and C are hidden (edge (AC) is not in the graph)**
  - they cannot sense their packet transmissions
- **Consequences for MAC protocols**
  - CSMA protocols will never increase CW
  - TDMA protocols will have to agree on a frame over multiple hops

Hidden terminal problem



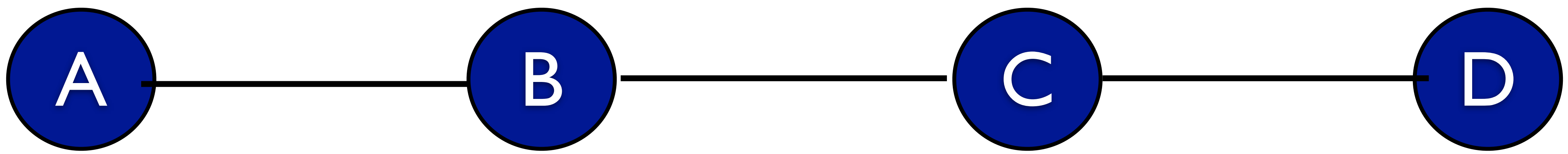
- **Node A and C are hidden (edge (AC) is not in the graph)**
  - they cannot sense their packet transmissions
- **Consequences for MAC protocols**
  - CSMA protocols will never increase CW
  - TDMA protocols will have to agree on a frame over multiple hops

Hidden terminal problem



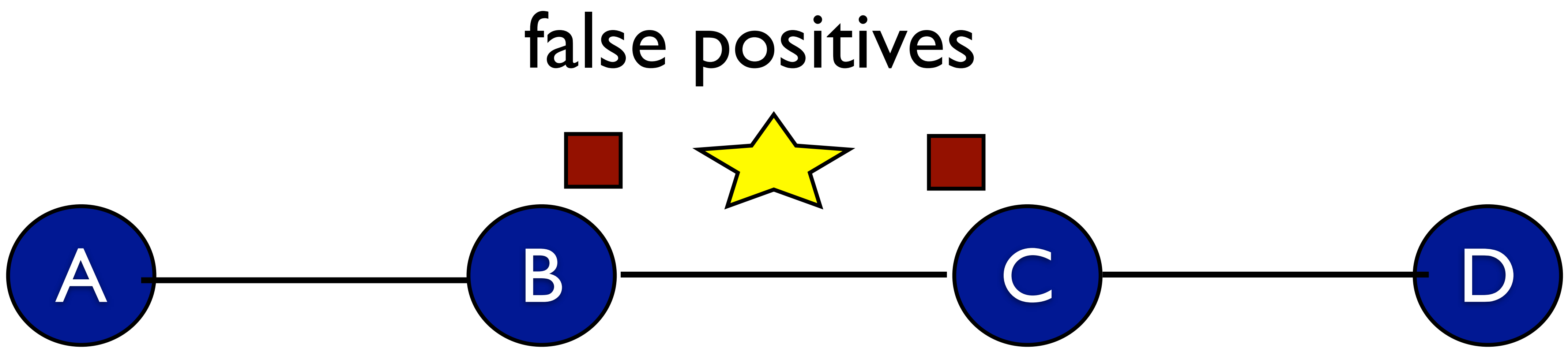
- **Node A and C are hidden (edge (AC) is not in the graph)**
  - they cannot sense their packet transmissions
- **Consequences for MAC protocols**
  - CSMA protocols will never increase CW
  - TDMA protocols will have to agree on a frame over multiple hops

Exposed terminal problem



- **Node B and C can communicate**
  - (BA) and (CD) can occur currently (collisions at receivers)
- **Consequences for MAC protocols**
  - CSMA will increase CW unnecessarily

Exposed terminal problem

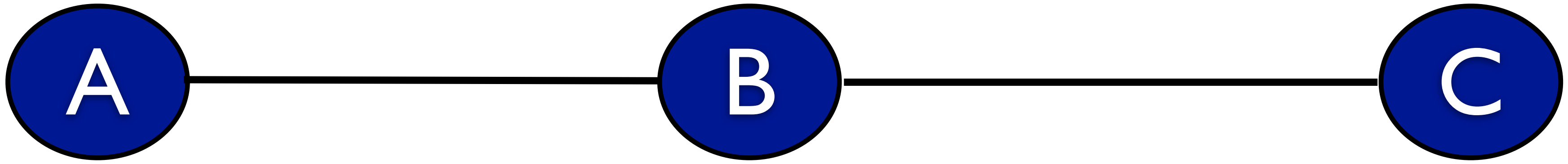


- **Node B and C can communicate**
  - (BA) and (CD) can occur currently (collisions at receivers)
- **Consequences for MAC protocols**
  - CSMA will increase CW unnecessarily

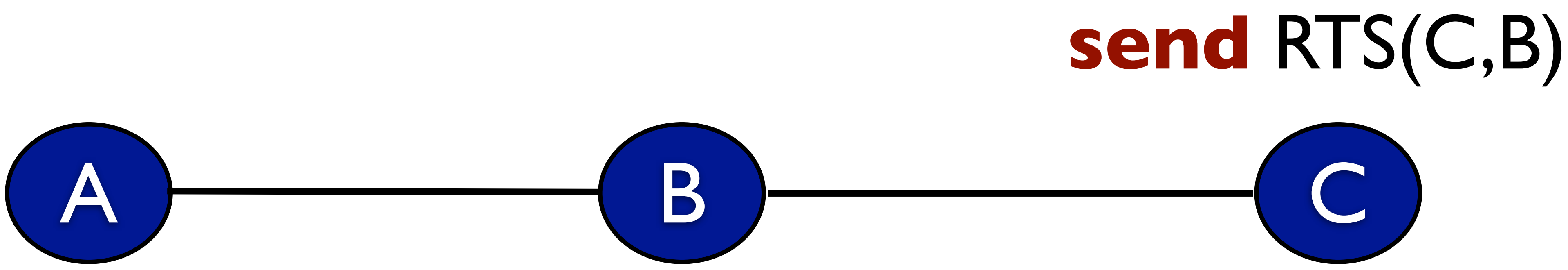
RTS/CTS a solution for CSMA protocols

- **Add two additional messages to the TDMA protocol**
  - RTS - request to send
  - CTS - clear to send
- **Algorithm**
  - node **n** wants to send packet to **m**
    - transmit **RTS**(n, m)
  - node a_1, a_2, ..., a_n, **m** receive **RTS**(n, m)
    - node **m** replies with **CTS**(n, m) if its channel is free
  - node b_1, b_2, ..., b_n, **n** receives **CTS**(n, m)
    - node **n** transmits the data packet
- **The algorithm signals access requests over 2-hops**

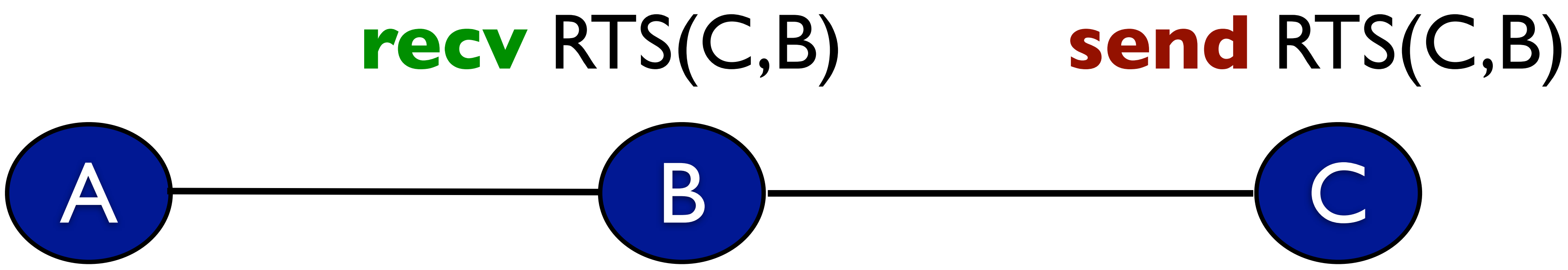
Hidden terminal problem



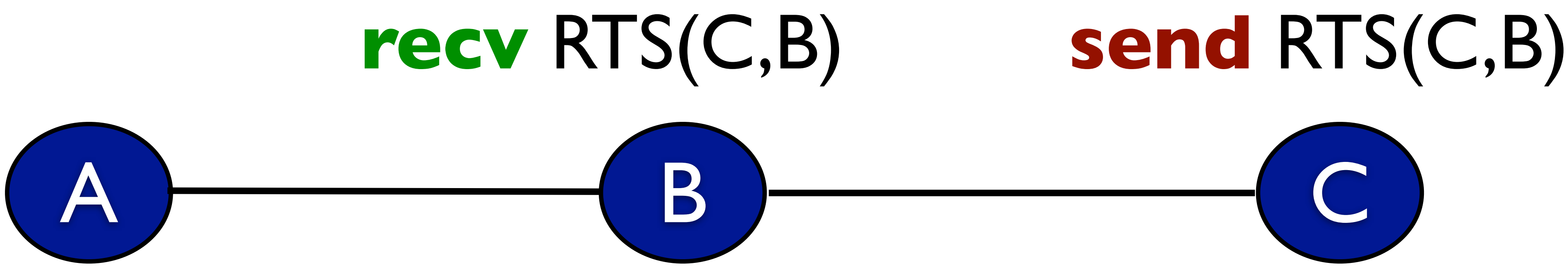
Hidden terminal problem

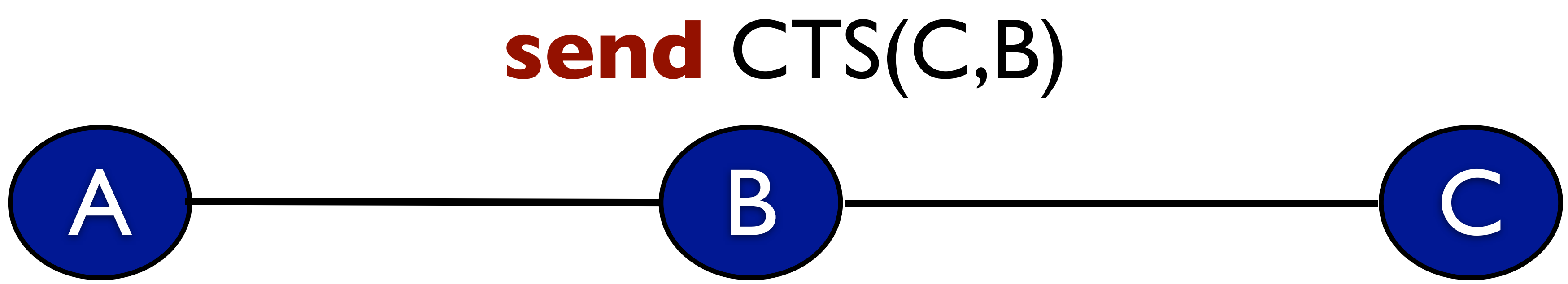


Hidden terminal problem

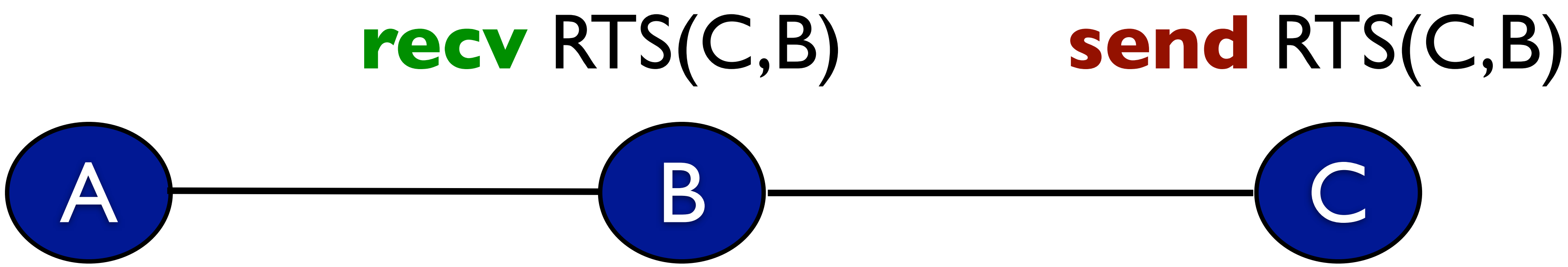


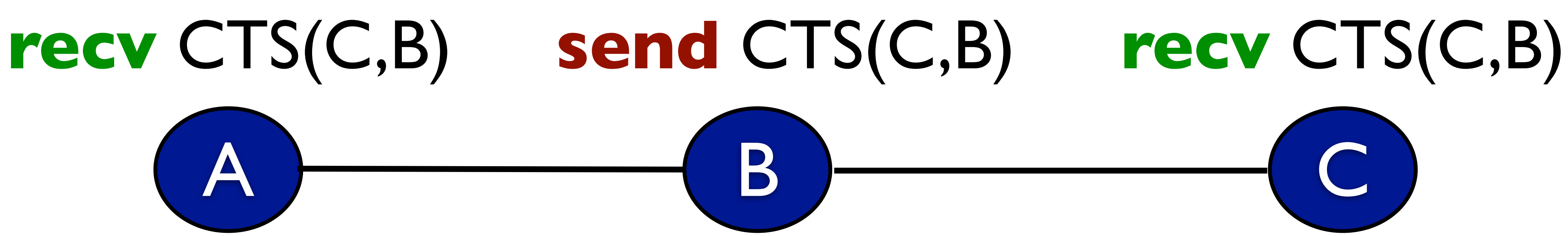
Hidden terminal problem





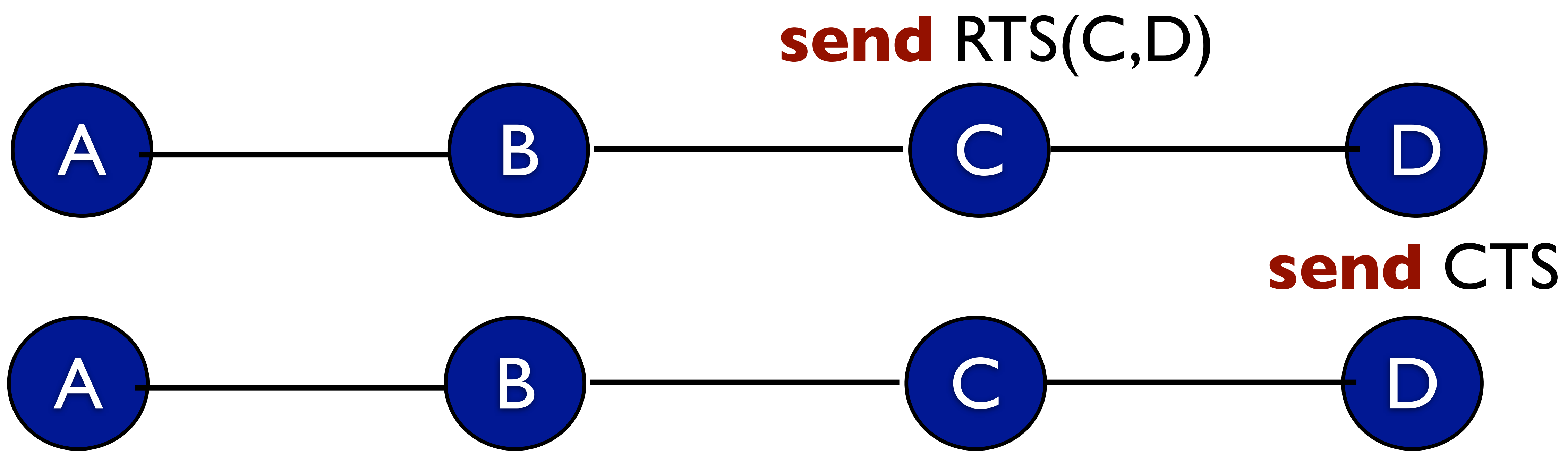
Hidden terminal problem



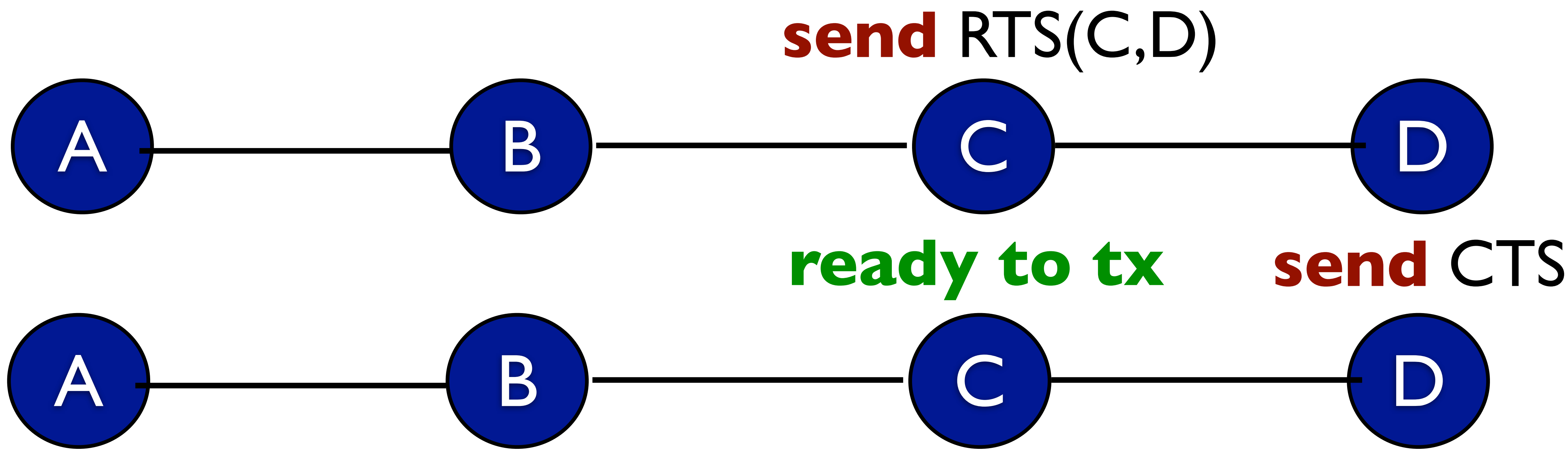


Exposed terminal problem

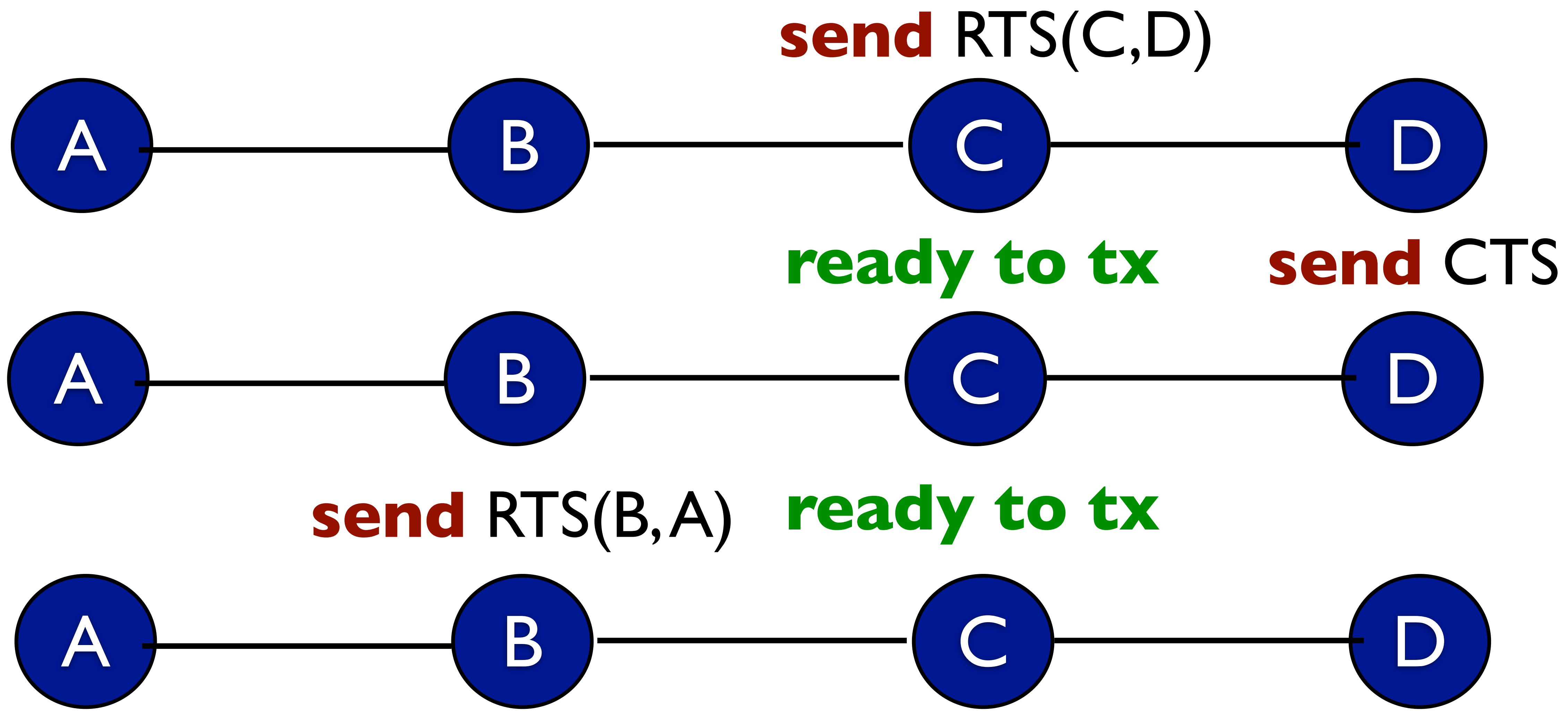
Exposed terminal problem



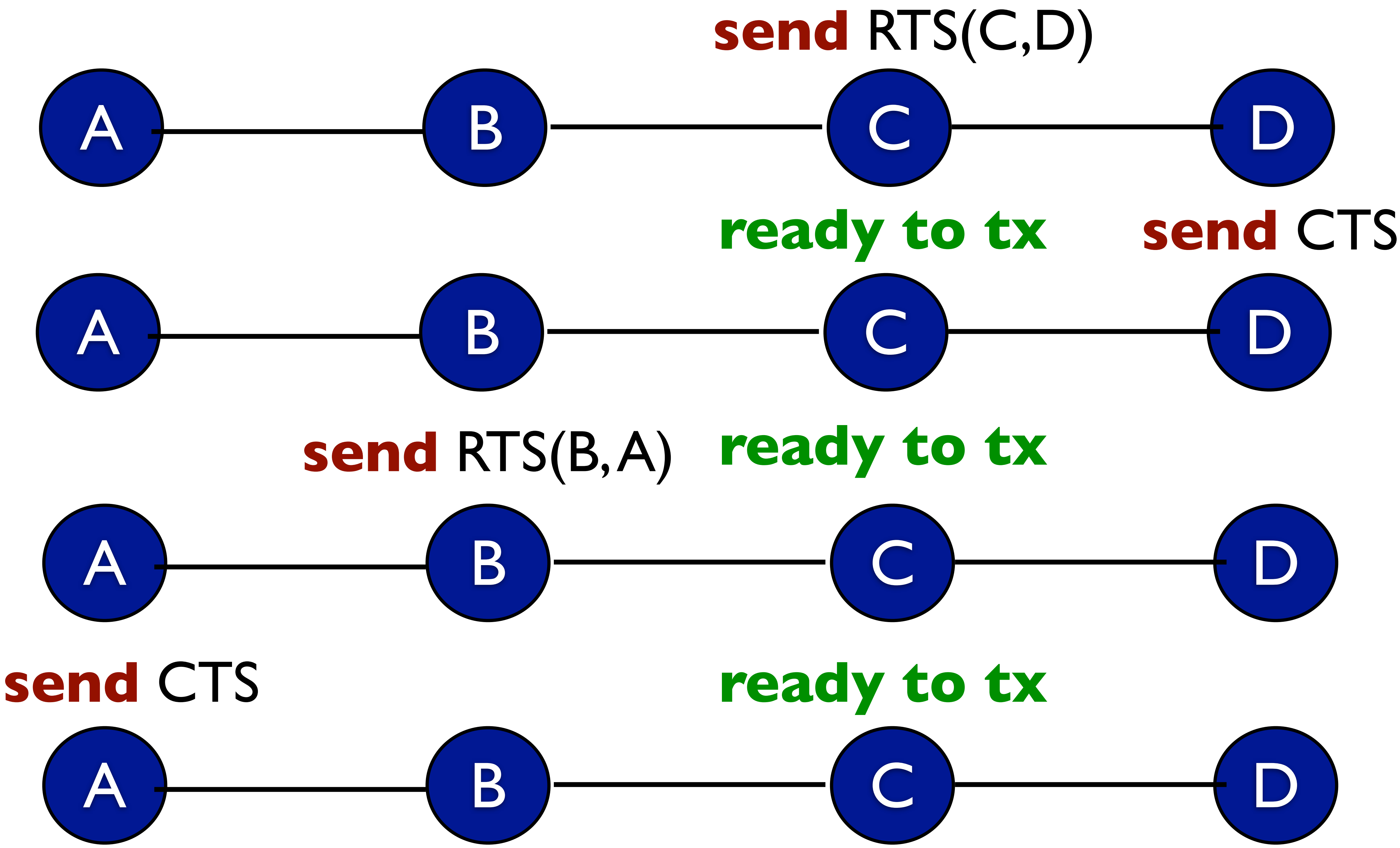
Exposed terminal problem



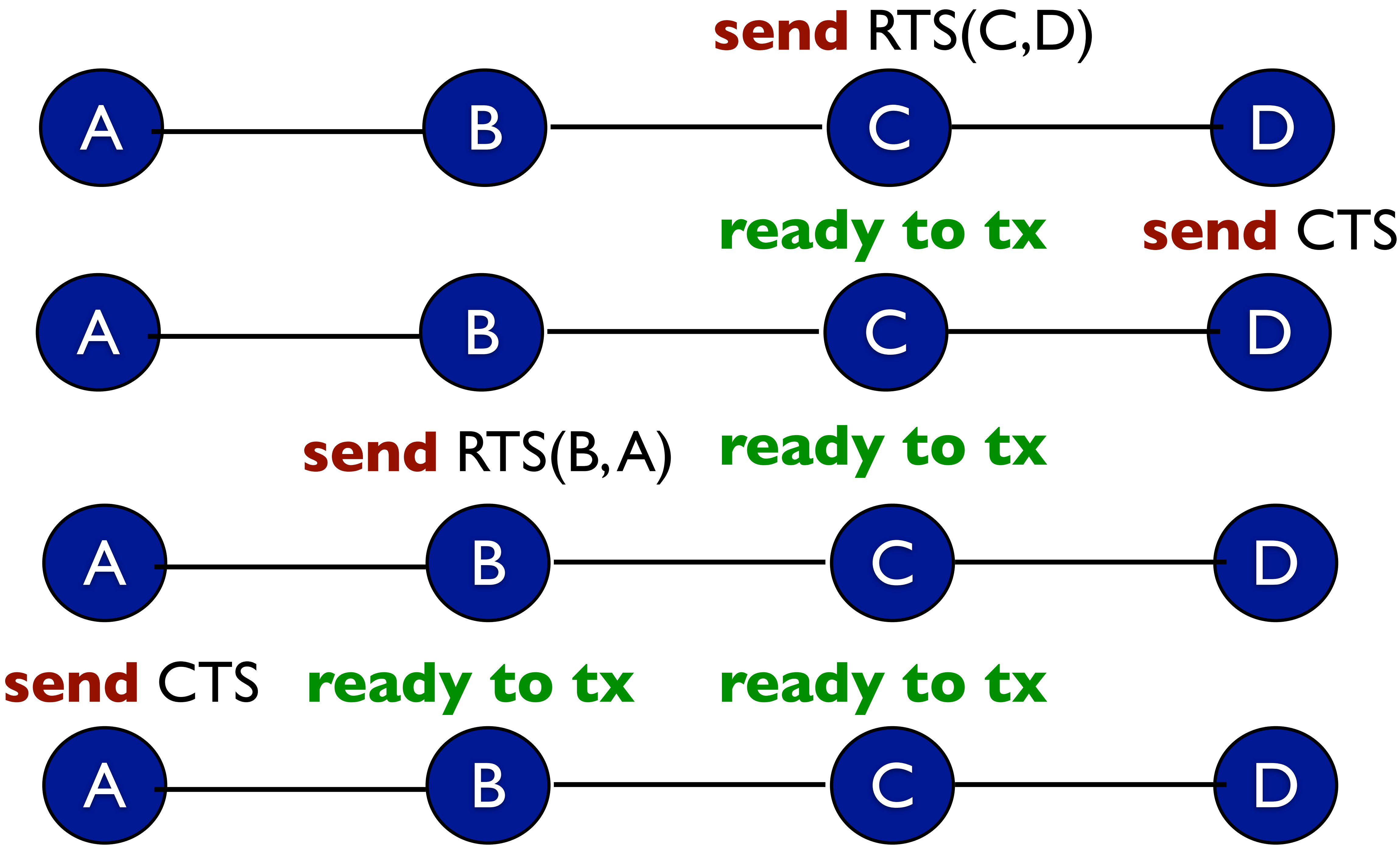
Exposed terminal problem



Exposed terminal problem



Exposed terminal problem



WLAN technology

- **Protocol soup:**

WiFi

Local wireless networks
WLAN 802.11

802.11a → 802.11h

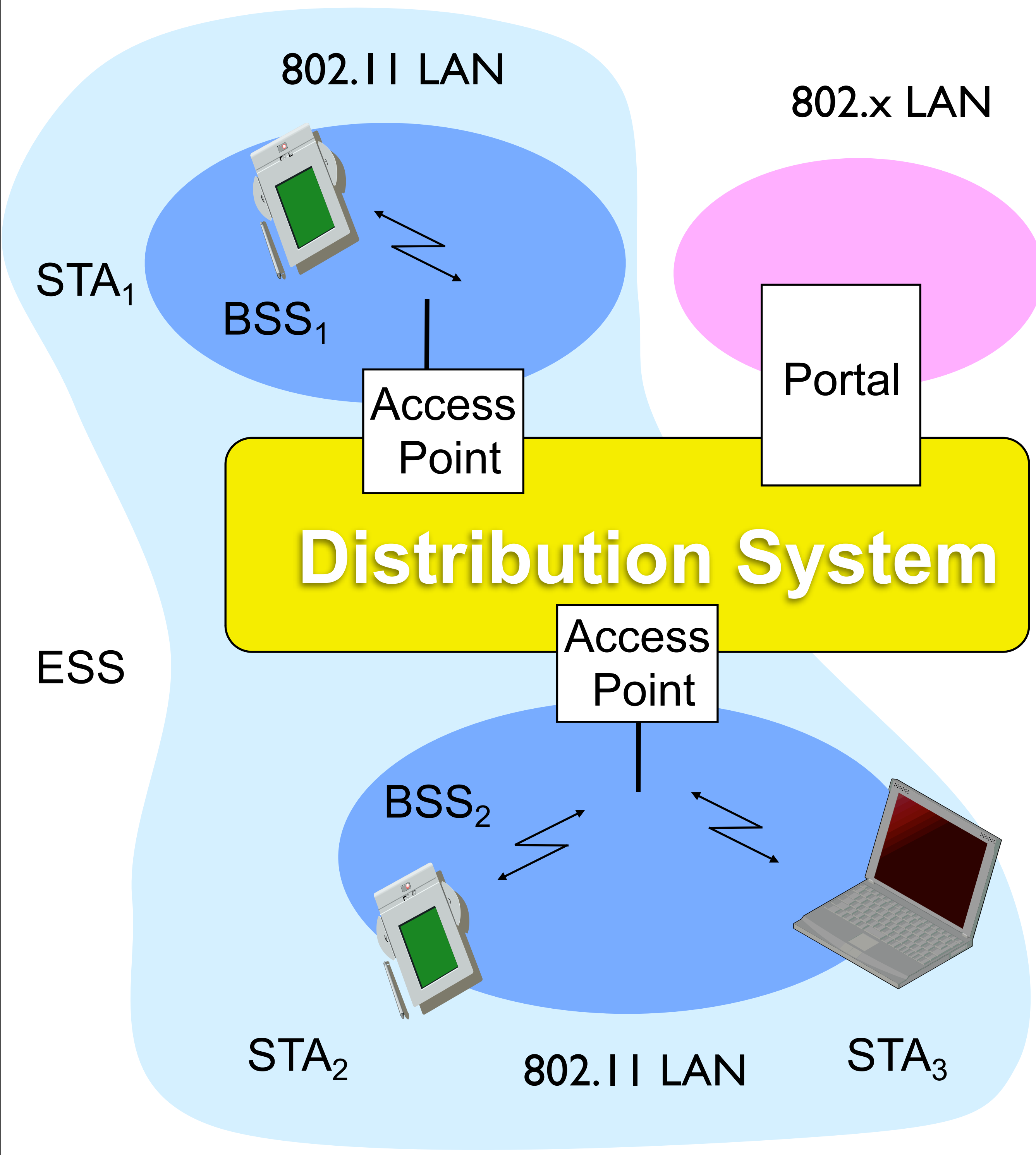
802.11i/e/…/w

802.11b → 802.11g

- **Goals:**
  - seamless operation
  - leverage on existing wired infrastructure
  - low-power operation on stations

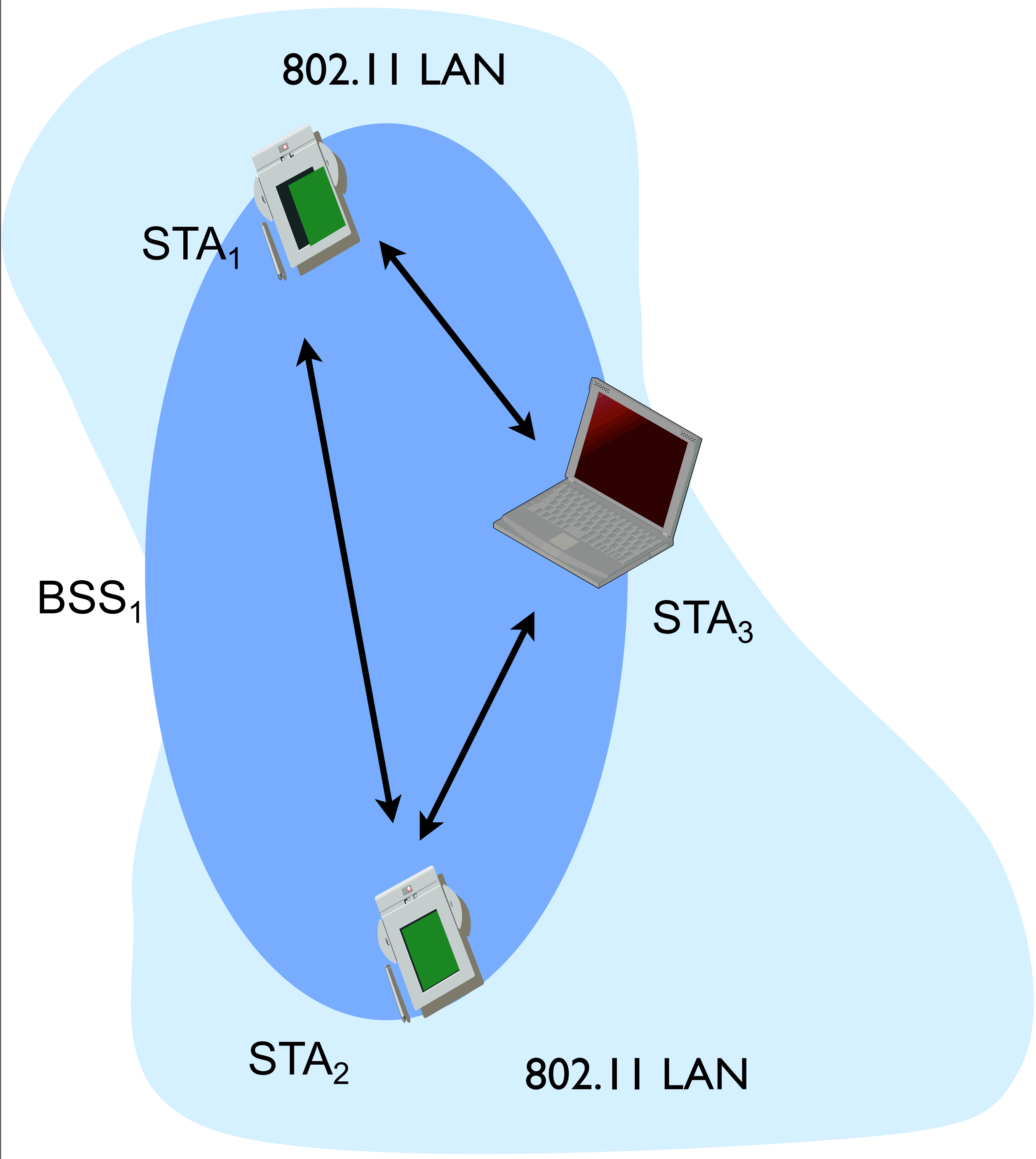
- **Two architectures: infrastructure + ad-hoc**

802.11: Architecture of an infrastructure network



- **Station (STA)**
  - terminal with wireless access mechanisms to contact the access point
- **Basic Service Set (BSS)**
  - group of stations using the same radio frequency
- **Access Point**
  - station integrated into the wireless LAN and the distribution system
- **Portal**
  - bridge to other (wired) networks
- **Distribution System**
  - interconnection network to/form one logical network

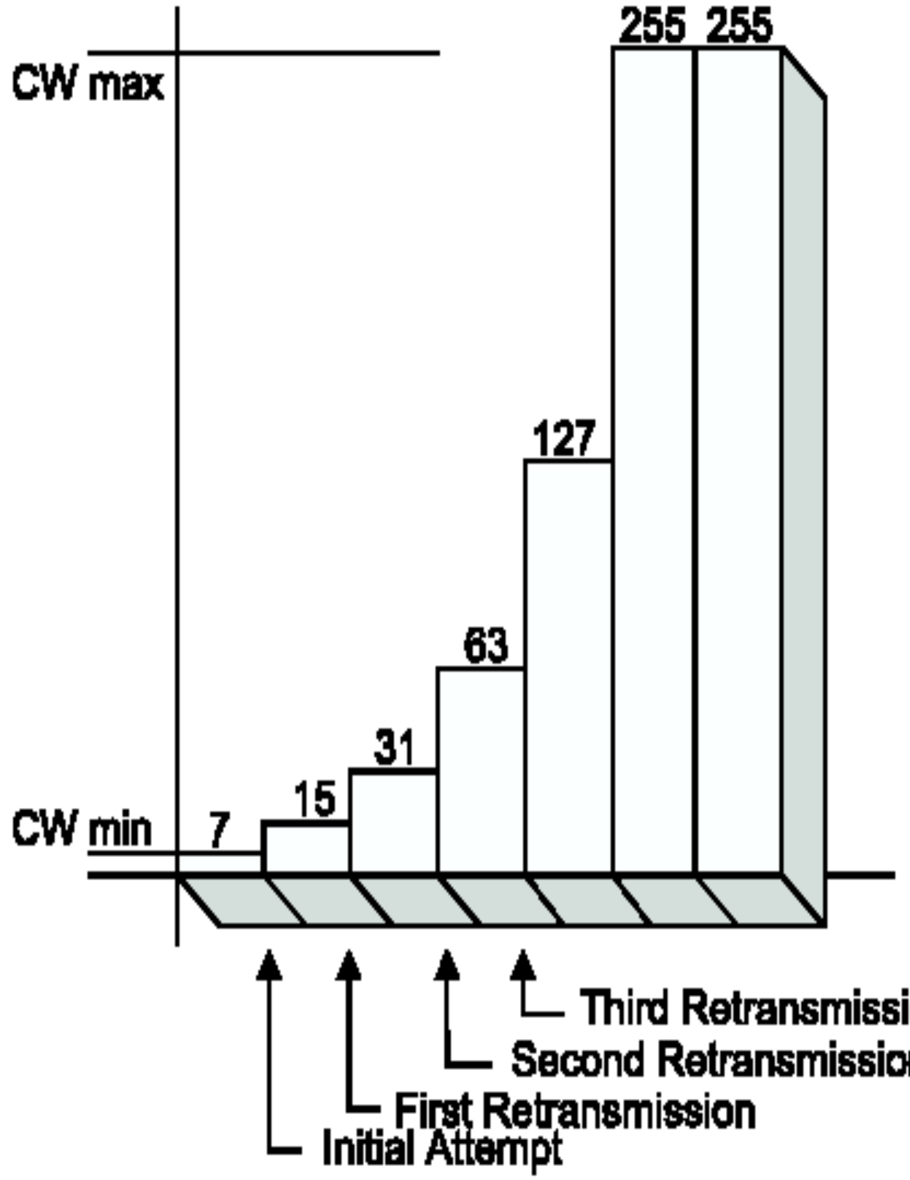
802.11: Architecture of an ad-hoc network



- **Direct communication within a limited range**
  - Station (STA): terminal with access mechanisms to the wireless medium
  - Independent Basic Service Set (IBSS): group of stations using the same radio frequency
- **When no direct link is feasible between two station, a third station may act as a relay (multi-hop communications)**

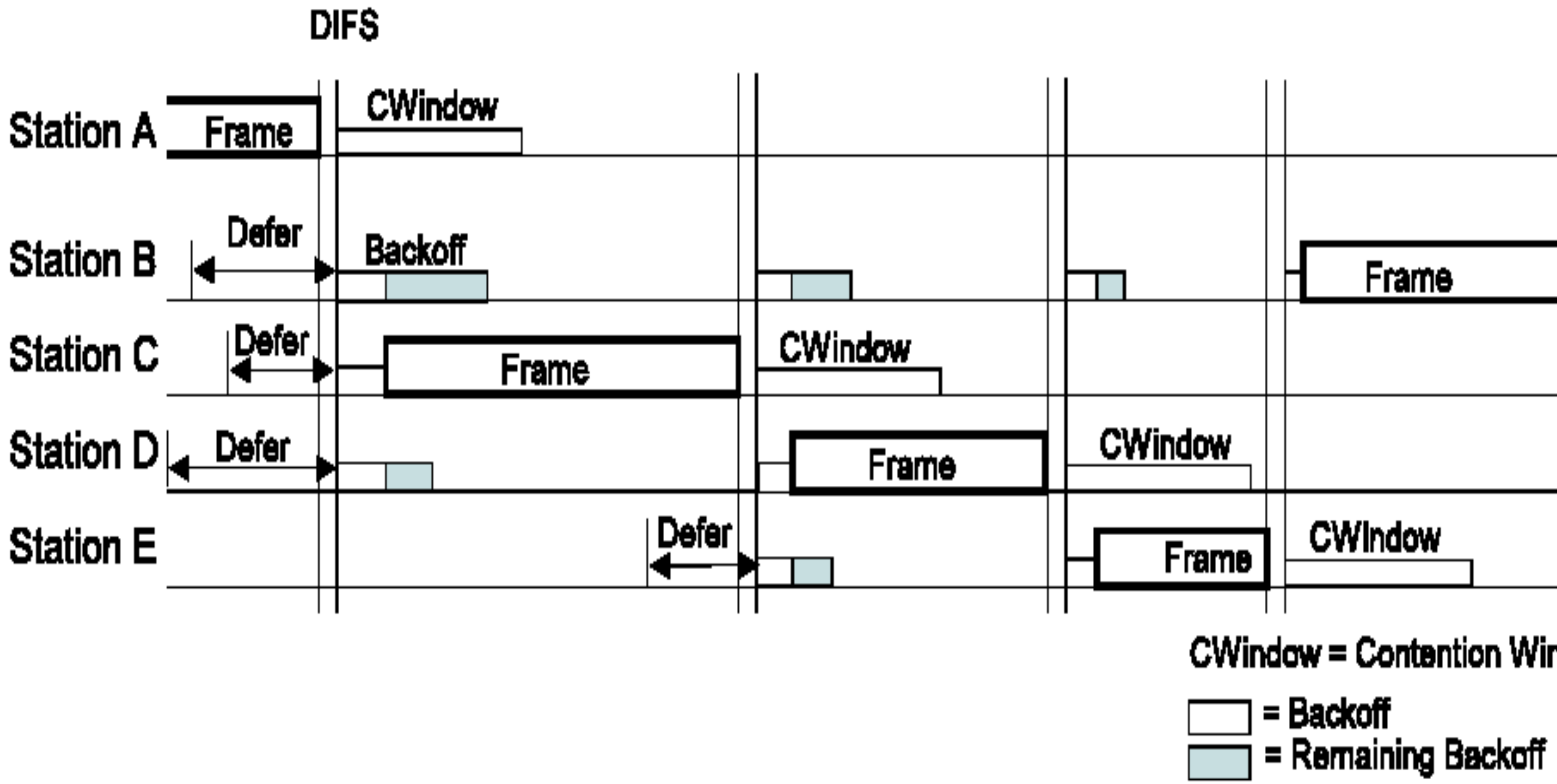
802.11b - Distributed Coordination Function

- **Exponential back-off**
  - Chosen for uniformly from (0, CW-1),
  - CW increase exponentially with the number of failed attempts
  - CW_{min} – minimum contention window
  - $CW_{max} = 2^m CW_{min}$ – maximum contention window



802.11b - Distributed Coordination Function

- Message resent when the backoff counter reaches zero
- Backoff counter decremented only when the channel is idle
- Backoff counter is reset to zero after a successful transmission



Routing

- **Routing consists of two fundamental steps**
  - Forwarding packets to the next hop (from an input interface to an output interface in a traditional wired network)
  - Determining how to forward packets (building a routing table or specifying a route)
- **Forwarding packets is easy, but knowing where to forward packets (especially efficiently) is hard**
  - Reach the destination
  - Minimize the number of hops (path length)
  - Minimize delay
  - Minimize packet loss
  - Minimize cost

Routing Decision Point

- **Source routing**
  - Sender determines a route and specifies it in the packet header
- **Hop-by-hop (datagram) routing**
  - A routing decision is made at each forwarding point (at each router)
  - Standard routing scheme for IP
- **Virtual circuit routing**
  - Determine and configure a path prior to sending first packet
  - Used in ATM (and analogous to voice telephone system)

Routing Table

- **A routing table contains information to determine how to forward packets**
  - Source routing: Routing table is used to determine route to the destination to be specified in the packet
  - Hop-by-hop routing: Routing table is used to determine the next hop for a given destination
  - Virtual circuit routing: Routing table used to determine path to configure through the network

Routing Approaches

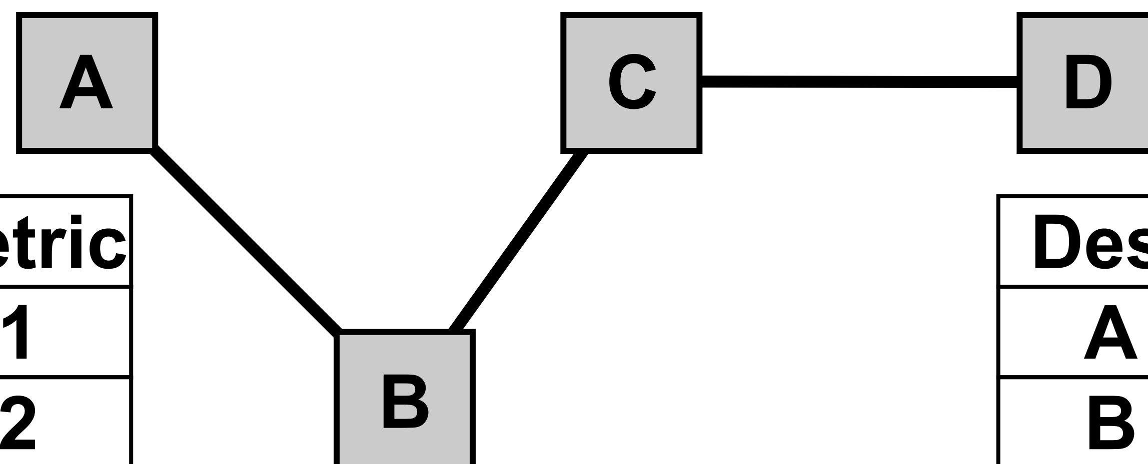
- **Reactive (On-demand) protocols**
  - discover routes when needed
  - source-initiated route discovery
- **Proactive protocols**
  - traditional distributed shortest-path protocols
  - based on periodic updates. High routing overhead
- **Tradeoff**
  - state maintenance traffic vs. route discovery traffic
  - route via maintained route vs. delay for route discovery

Distance Vector Algorithms (1)

- **"Distance" of each link in the network is a metric that is to be minimized**
  - each link may have "distance" 1 to minimize hop count
  - algorithm attempts to minimize distance
- **The routing table at each node…**
  - specifies the next hop for each destination
  - specifies the distance to that destination
- **Neighbors can exchange routing table information to find a route (or a better route) to a destination**

Distance Vector Algorithms (2)

Dest	Next	Metric
A	B	2
B	B	1
D	D	1



Dest	Next	Metric
B	B	1
C	B	2
D	B	3

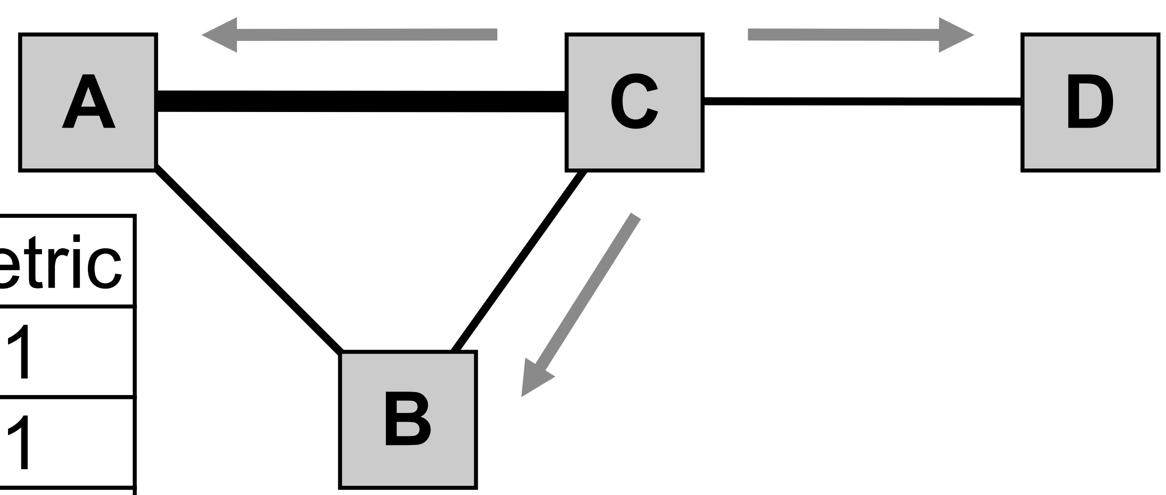
Dest	Next	Metric
A	A	1
C	C	1
D	C	2

Dest	Next	Metric
A	C	3
B	C	2
C	C	1

Distance Vector Algorithms (3)

- **Node A will learn of Node C's shorter path to Node D and update its routing table**

Dest	Next	Metric
A	A	1
B	B	1
D	D	1

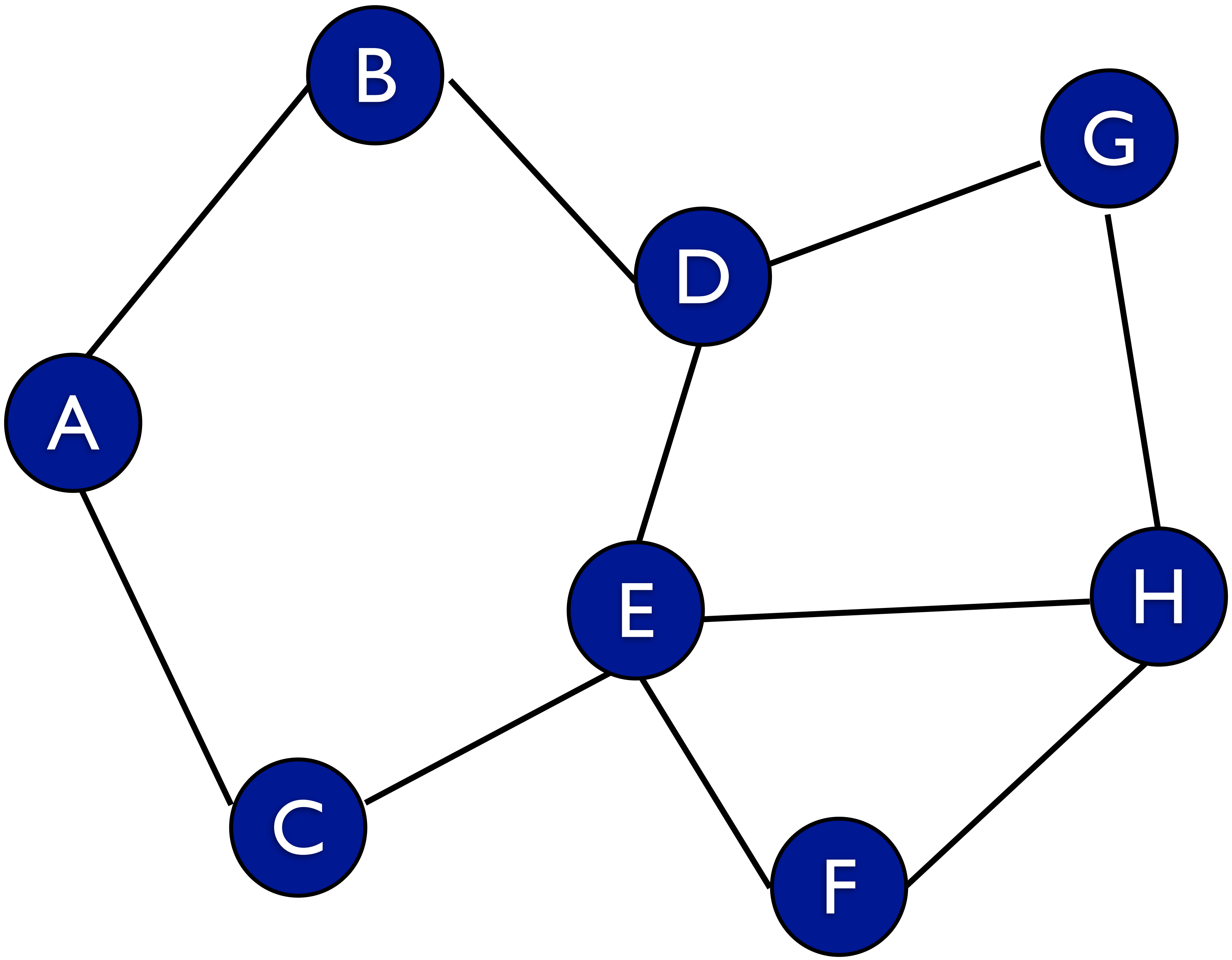


Dest	Next	Metric
B	B	1
C	C	1
D	C	2

Reactive Routing – Source initiated

- **Source floods the network with a route request packet when a route is required to a destination**
  - flood is propagated outwards from the source
  - pure flooding = every node transmits the request only once
- **Destination replies to request**
  - reply uses reversed path of route request
  - sets up the forward path

Route Discovery



RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route

A-B-C

Route Request (RREQ)

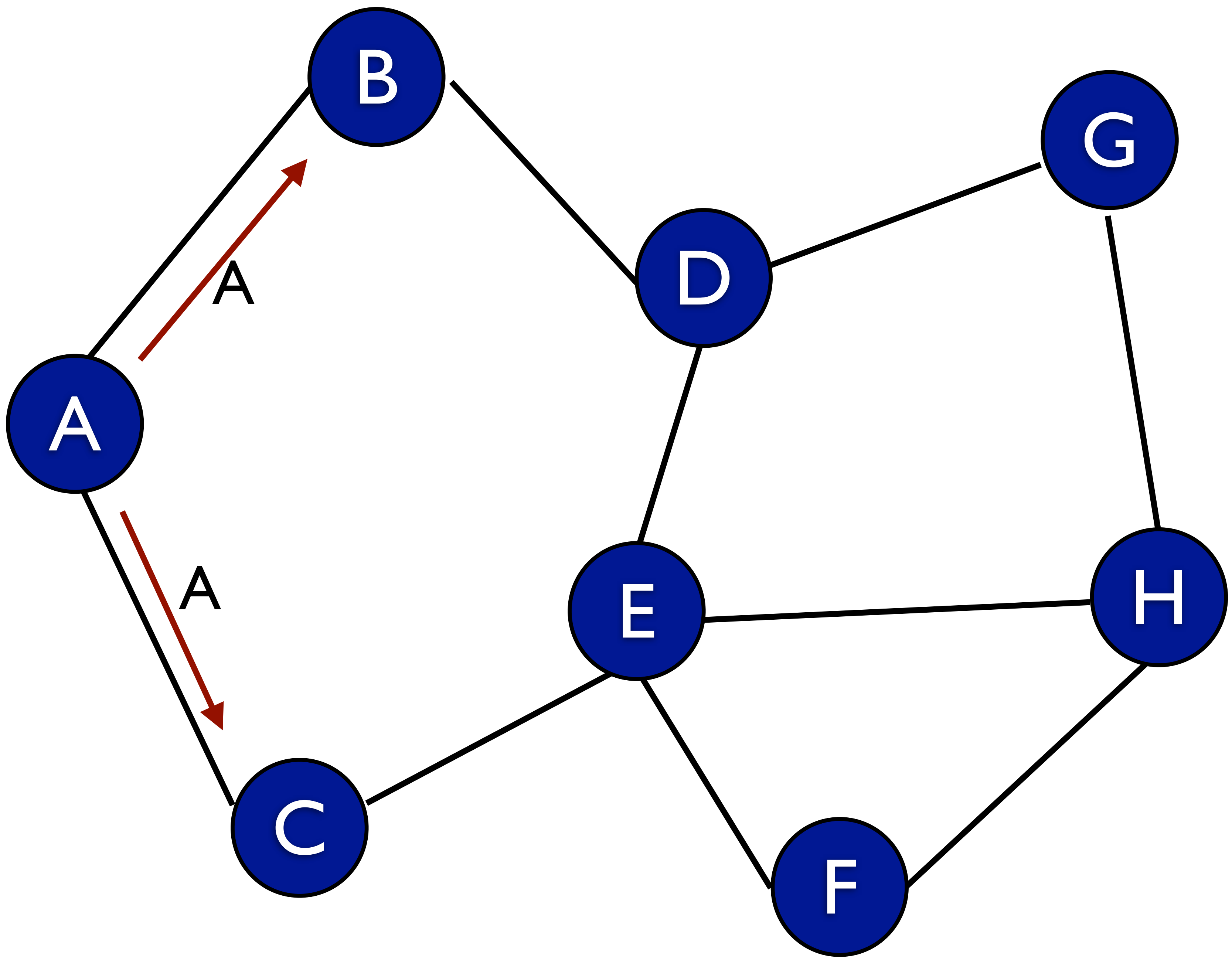
A-B-C

Route Reply (RREP)

Route Discovery

RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route



A-B-C

Route Request (RREQ)

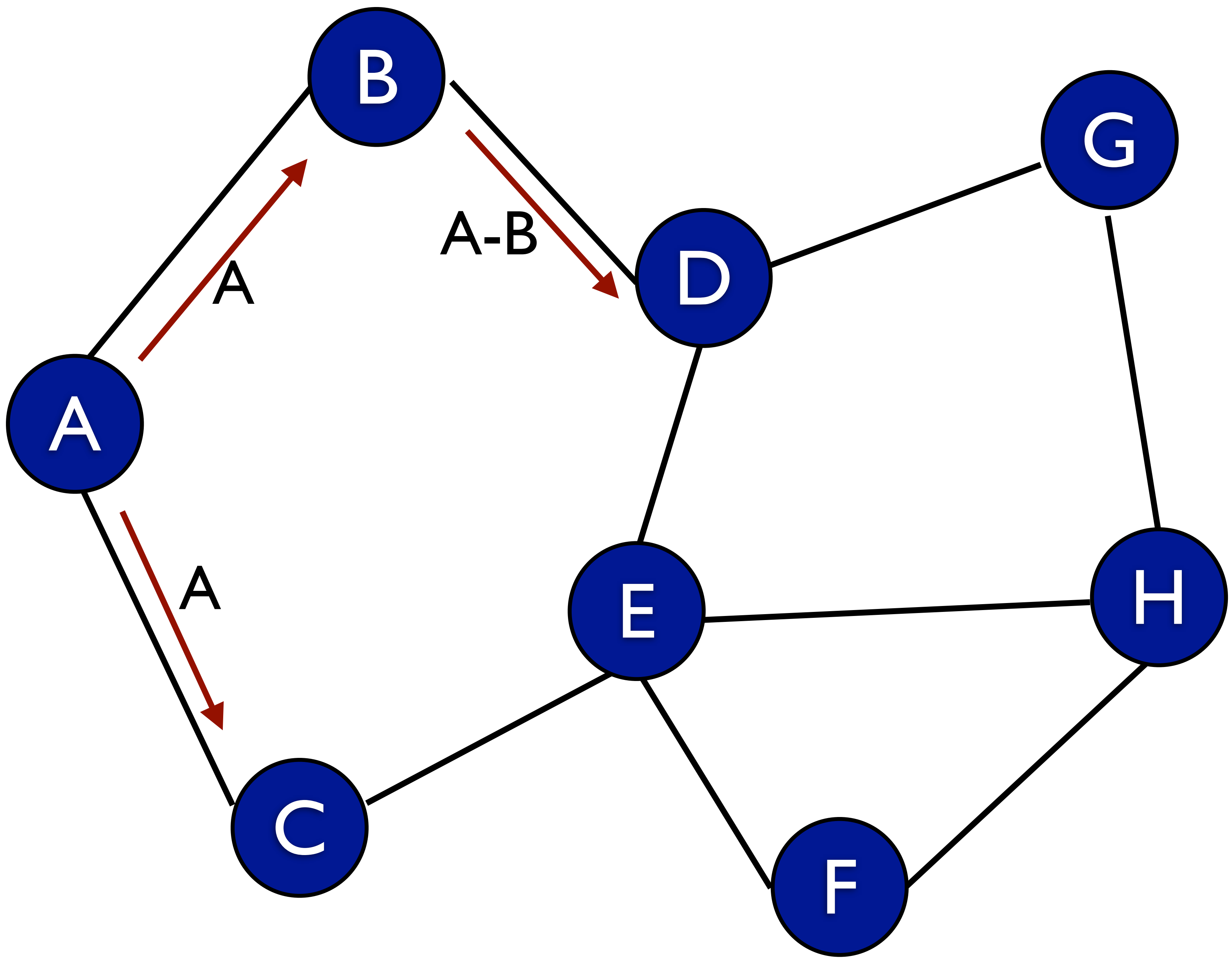
A-B-C

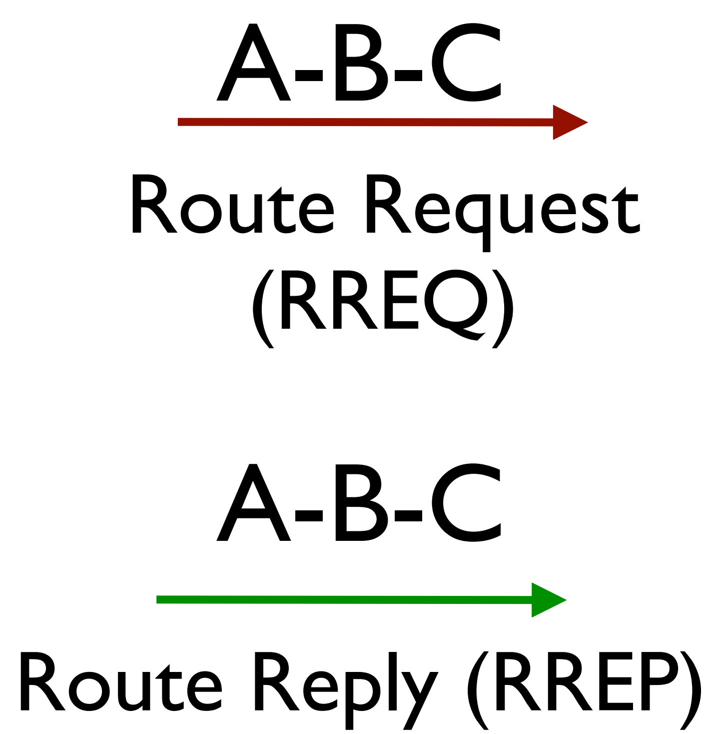
Route Reply (RREP)

Route Discovery

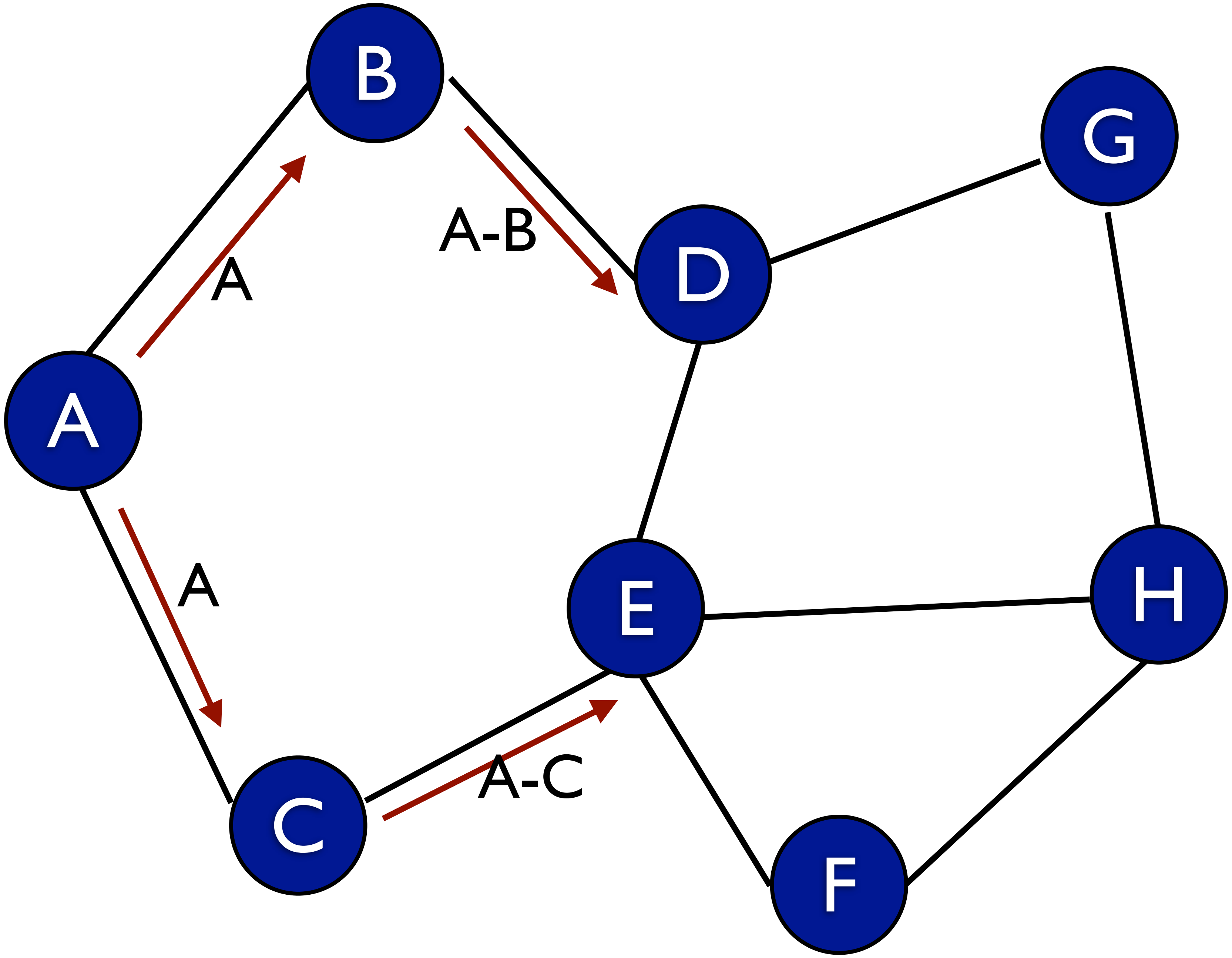
RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route



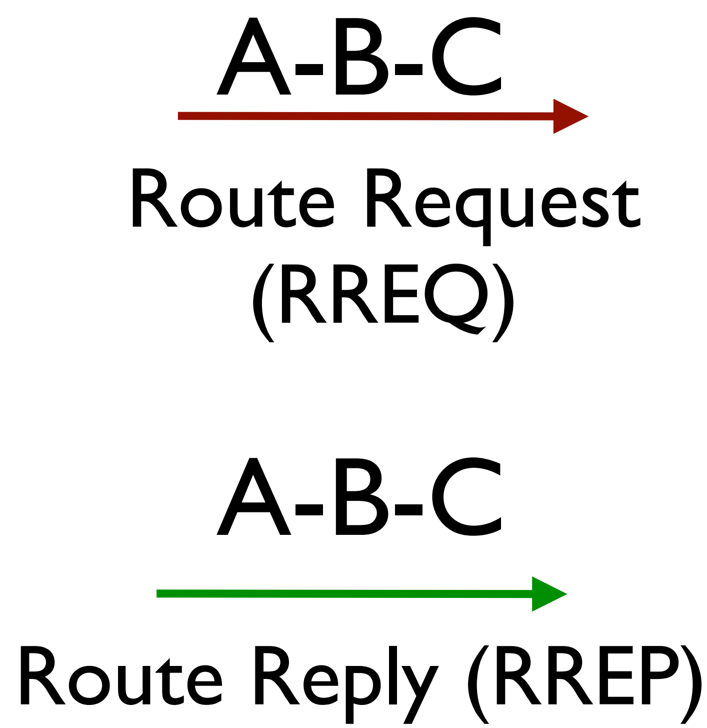


Route Discovery

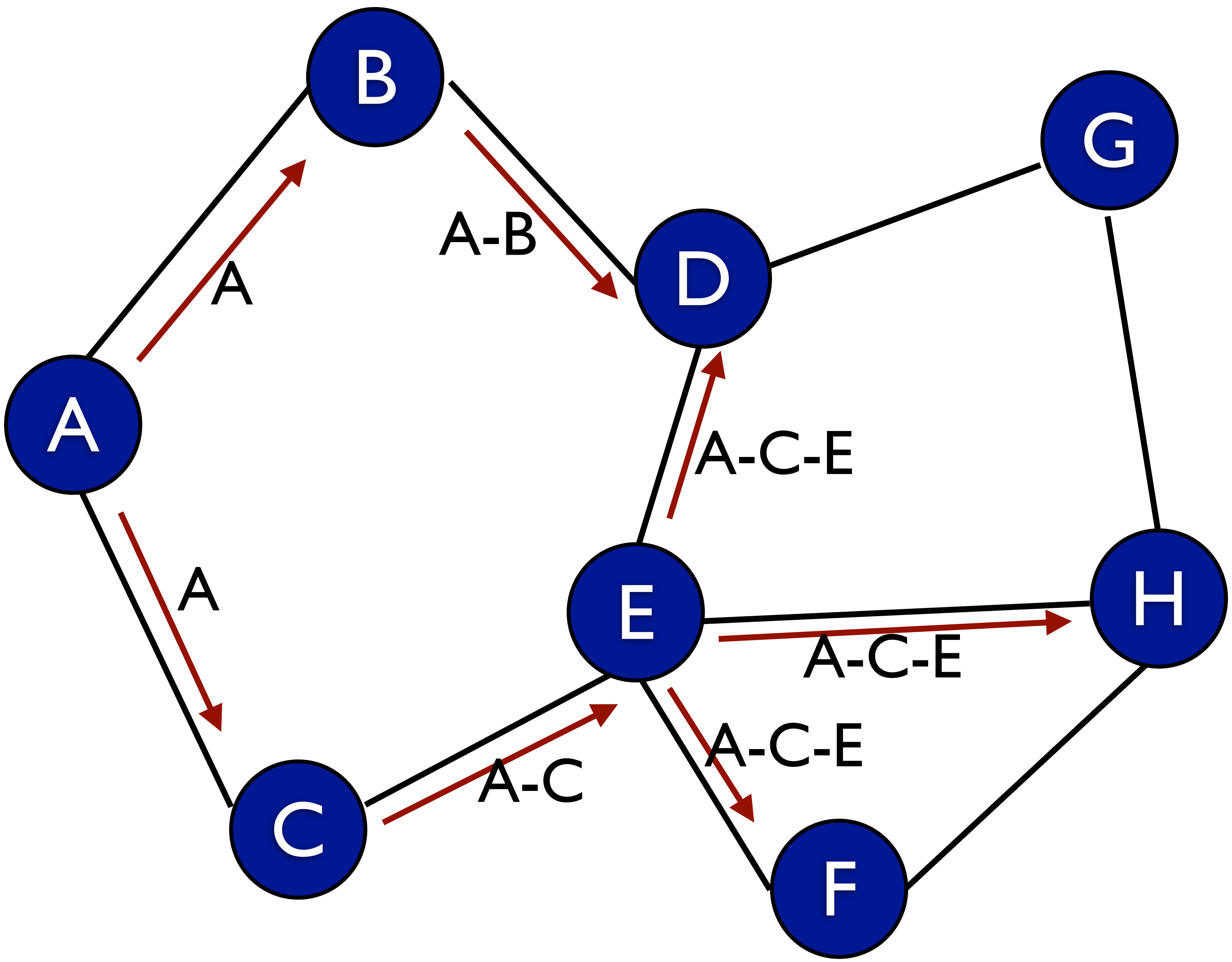


RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route



Route Discovery



RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route

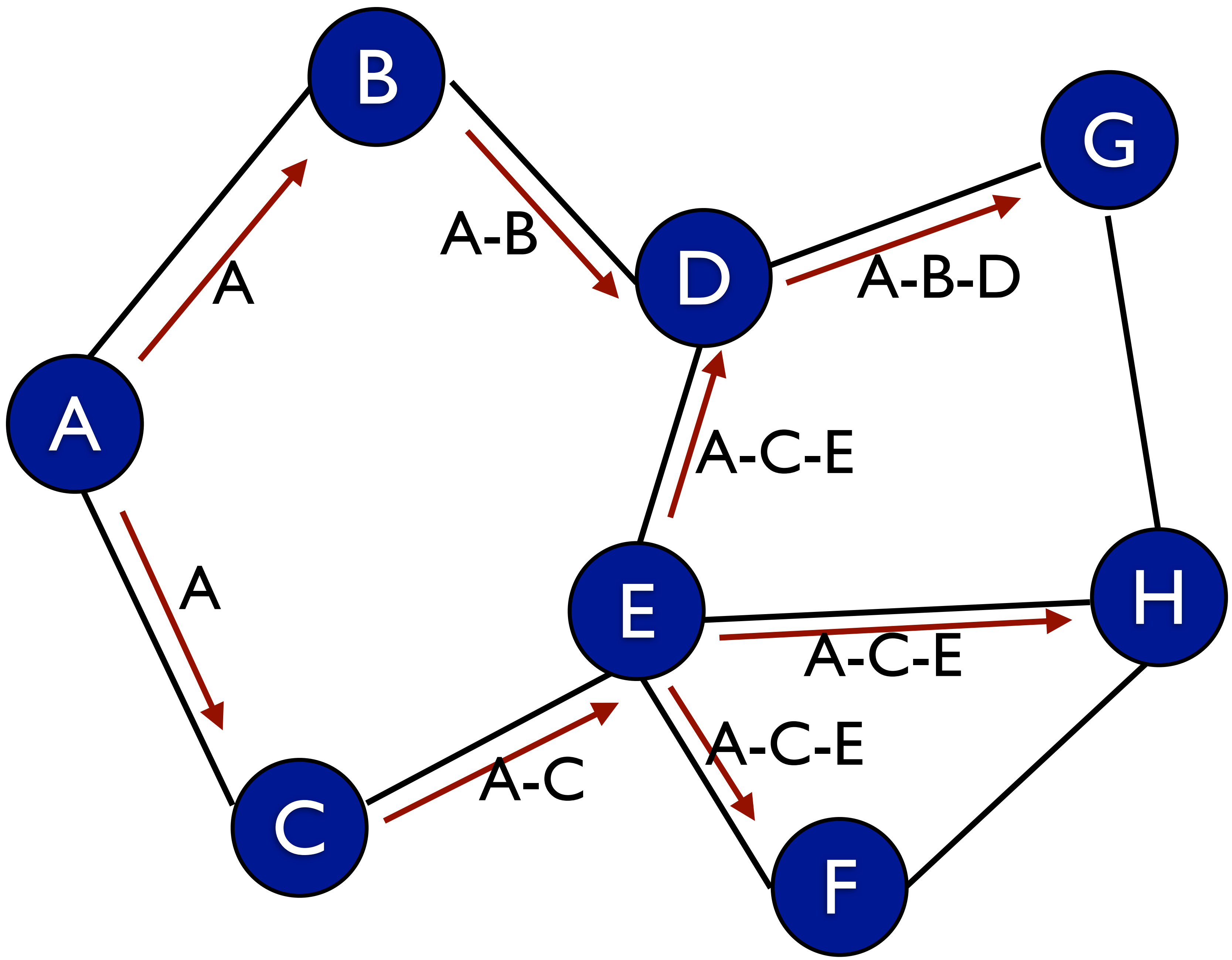
A-B-C

Route Request (RREQ)

A-B-C

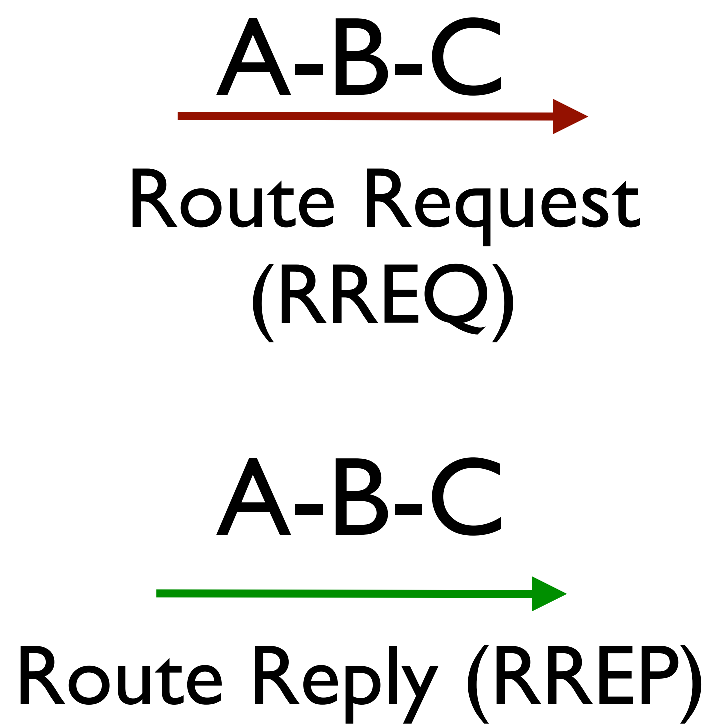
Route Reply (RREP)

Route Discovery

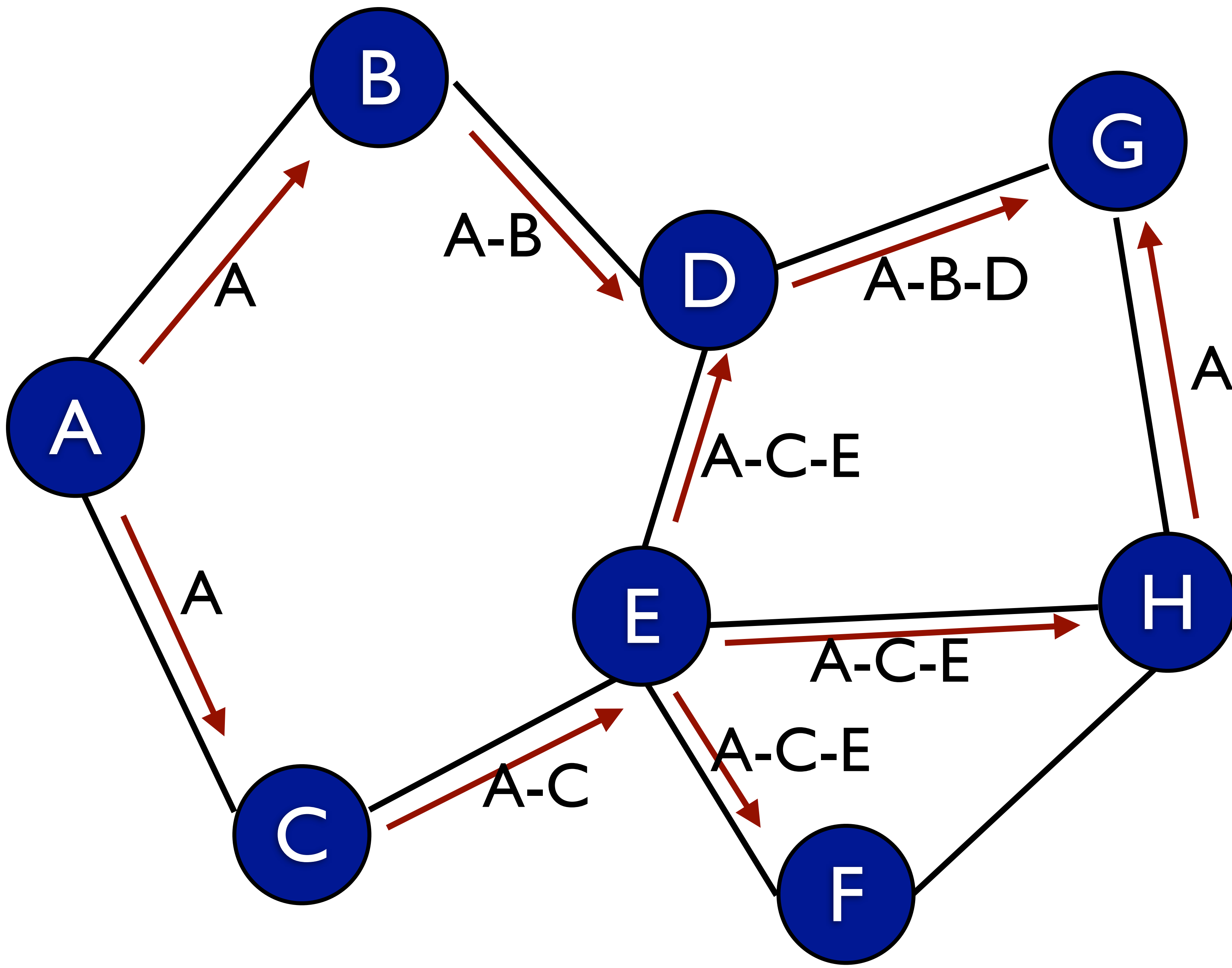


RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route



Route Discovery



RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route

A-B-C

Route Request (RREQ)

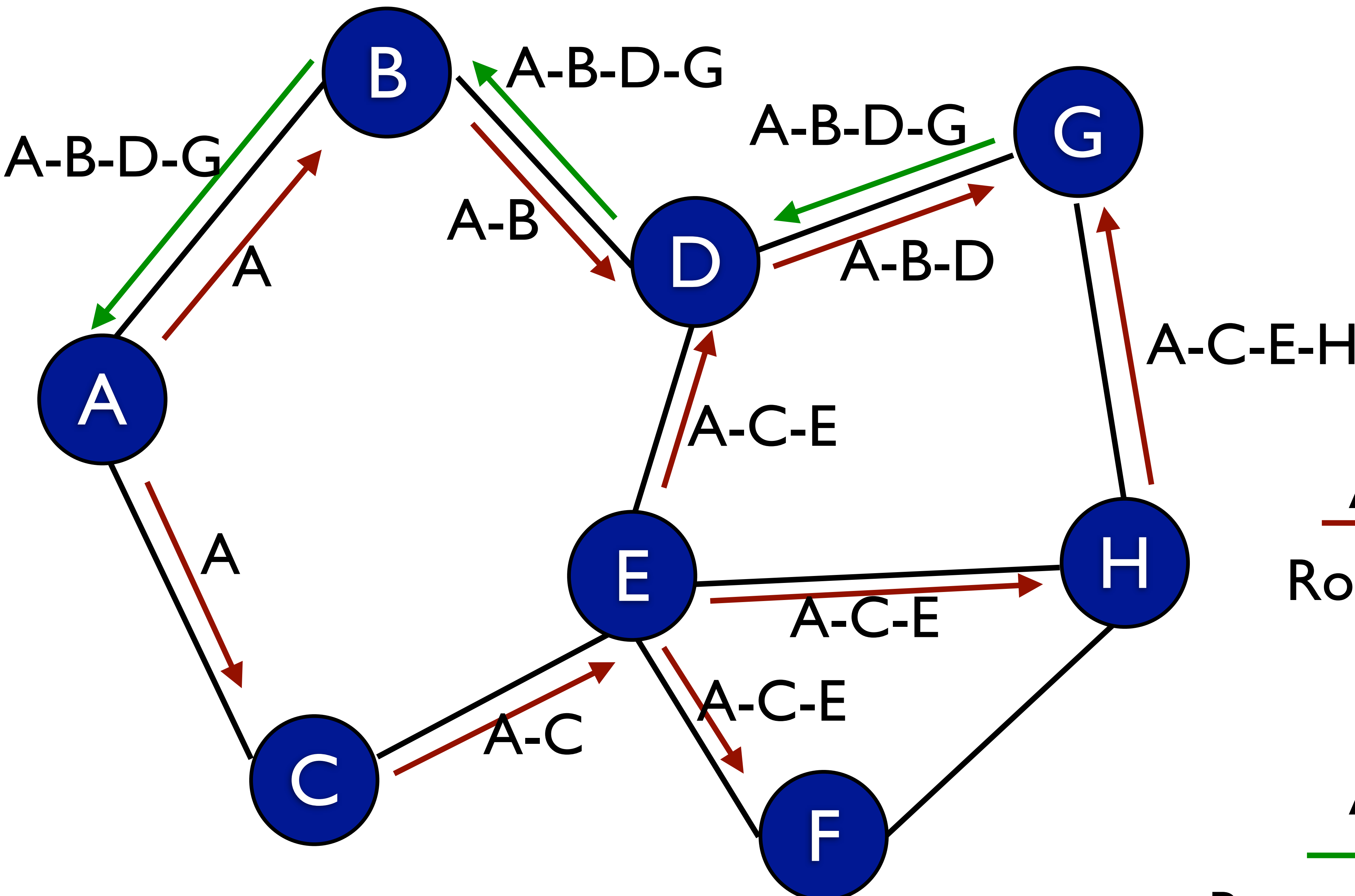
A-B-C

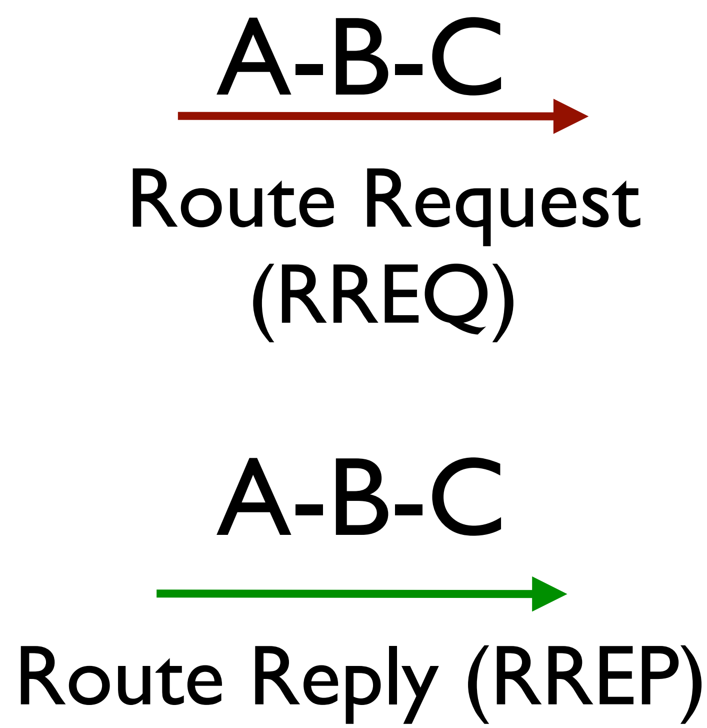
Route Reply (RREP)

Route Discovery

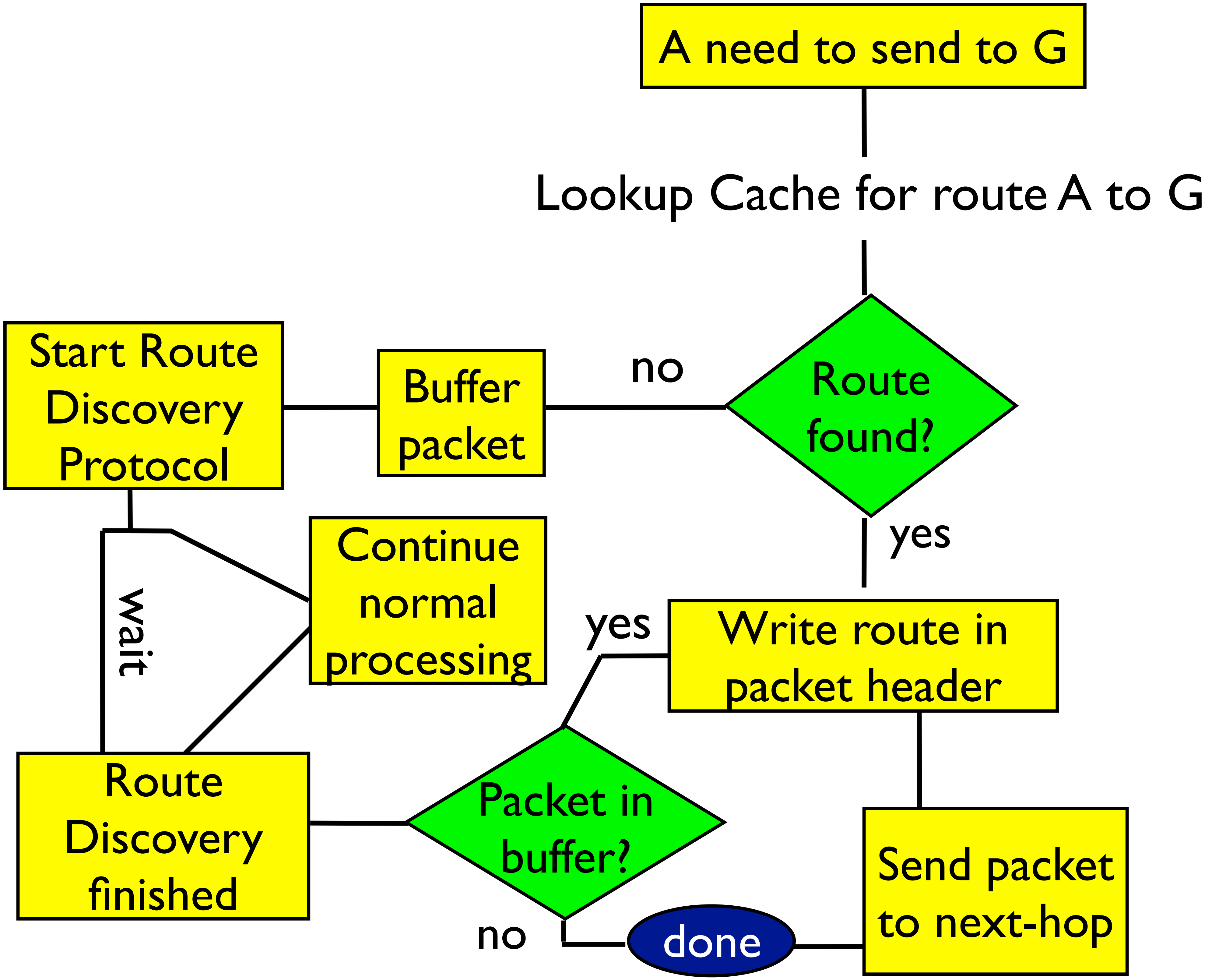
RREQ FORMAT

Initiator
Initiator seq #
Destination
Partial Route

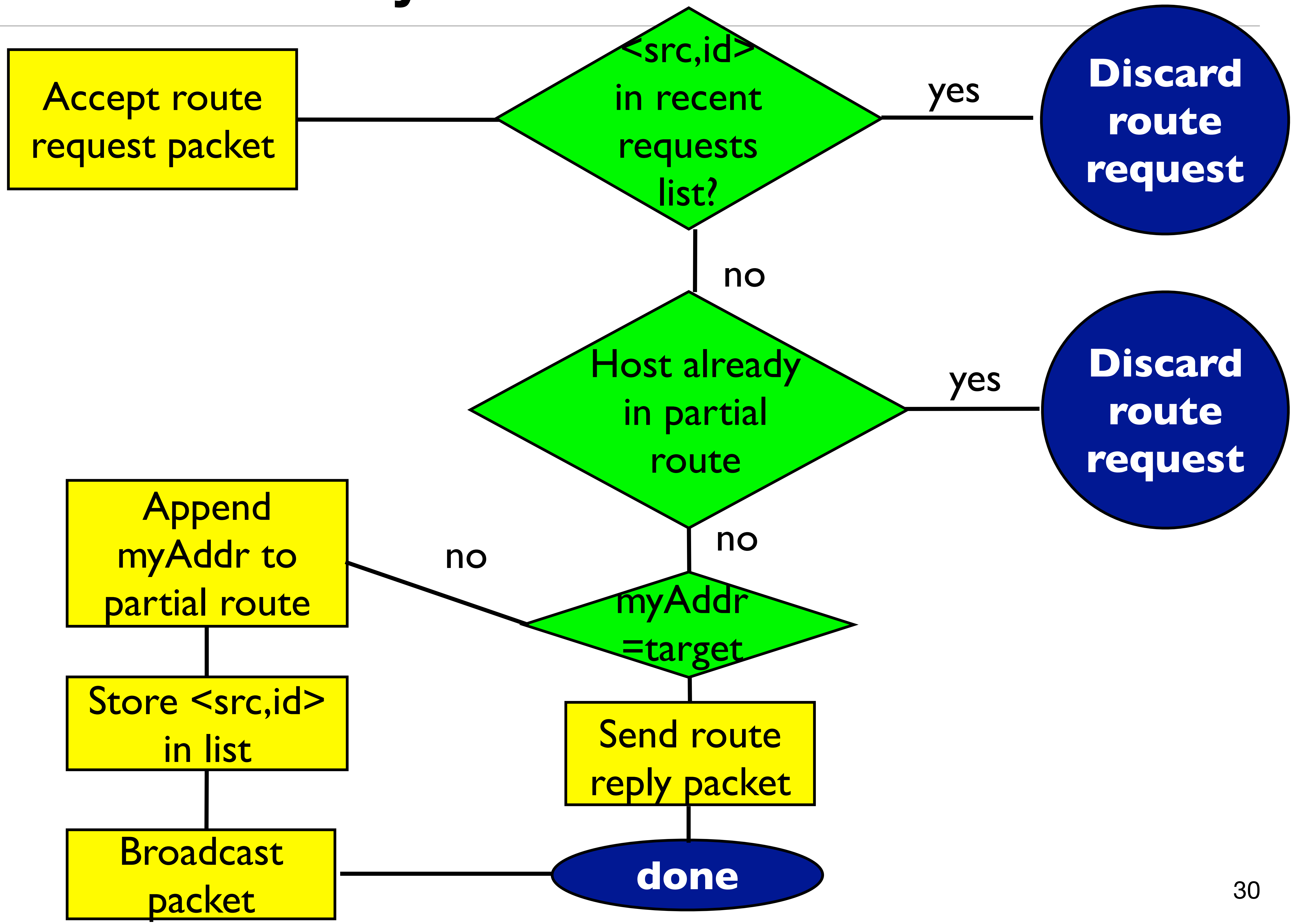




Route Discovery: at source A



Route Discovery: At an intermediate node



Route Discovery

- **Route Reply message containing path information is sent back to the source either by**
  - the destination, or
  - intermediate nodes that have a route to the destination
  - reverse the order of the route record, and include it in Route Reply.
  - unicast, source routing
- **Each node maintains a Route Cache which records routes it has learned and overheard over time**

Route Maintenance

- **Route maintenance performed only while route is in use**
- **Error detection:**
  - monitors the validity of existing routes by passively listening to data packets transmitted at neighboring nodes
- **When problem detected, send Route Error packet to original sender to perform new route discovery**
  - Host detects the error and the host it was attempting;
  - Route Error is sent back to the sender the packet – original src