



ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithmen für drahtlose Netzwerke

Flooding und DSR

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer



Besonderheiten von Routing in Mobilen Ad-Hoc-Netzwerken

- ▶ **Mobilität der Teilnehmer**
 - Verbindungsverlust und Verbindungsaufbau ist häufiger als in anderen drahtlosen Netzwerken
 - Besonders bei hoher Geschwindigkeit
- ▶ **Andere Performanz-Kriterien**
 - Routen-Stabilität trotz Mobilität
 - Energieverbrauch

Unicast Routing

- ▶ **Vielzahl von Protokollen untersucht**
 - Adaptionen und Neuentwicklungen
- ▶ **Kein Protokoll dominiert die anderen in allen Situationen**
 - Lösung: adaptive Protokolle?

Routing in MANETs

- ▶ **Routing**
 - Bestimmung der Wege
 - Transport der Daten
- ▶ **Protokollarten**
 - Proaktiv
 - Routing-Tabellen mit Aktualisierungen
 - Reaktiv
 - Bestimmung oder Aktualisierung der Wege nur bei Bedarf
 - Hybrid
 - Mischform aus Proaktiv und Reaktive

Routing Protocols

▶ Proaktiv

- Routenbestimmung **unabhängig von Bedarf**
- Standard Link-State und Distance-Vector-Protokolle
 - Destination Sequenced Distance Vector (**DSDV**)
 - Optimized Link State Routing (**OLSR**)

▶ Reaktiv

- Routen werden erst bei **Bedarf** ermittelt
 - Dynamic Source Routing (**DSR**)
 - Ad hoc On-demand Distance Vector (**AODV**)
 - Dynamic MANET On-demand Routing Protocol
 - Temporally Ordered Routing Algorithm (**TORA**)

▶ Hybrid

- Kombination aus reaktiv und proaktiv
 - Zone Routing Protocol (**ZRP**)
 - Greedy Perimeter Stateless Routing (**GPSR**)

Trade-Off

- ▶ **Latenz durch der Routen-Suche**
 - Proaktive Protokolle sind schneller
 - Reaktive Protokolle müssen erst Routen bestimmen
- ▶ **Overhead durch Routen-Suche und Unterhalt**
 - Reaktive Protokolle haben geringen Overhead (Anzahl Nachrichten)
 - Proaktive Protokolle können höheren Aufwand haben
- ▶ **Traffic-Pattern und Mobilität**
 - entscheidend welcher Protokolltyp effizienter ist

Flooding

▶ **Algorithmus**

- Sender S broadcastet Datenpaket an alle Nachbarn
- Jeder Knoten der ein neues Paket erhält und nicht der Empfänger ist
 - broadcastet das Paket an alle Nachbarn

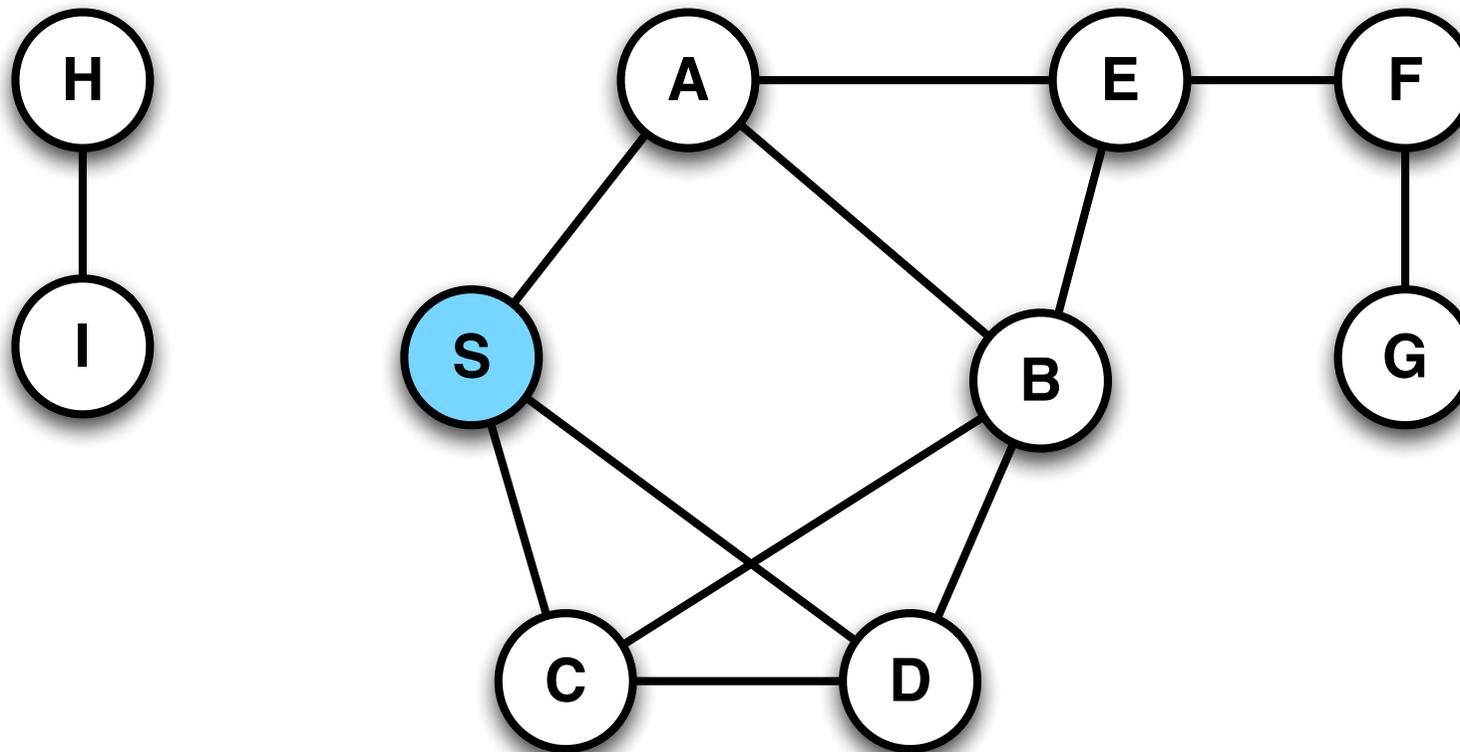
▶ **Sequenznummern**

- identifizieren die Nachrichten, damit keine Nachrichten mehrfach versendet werden

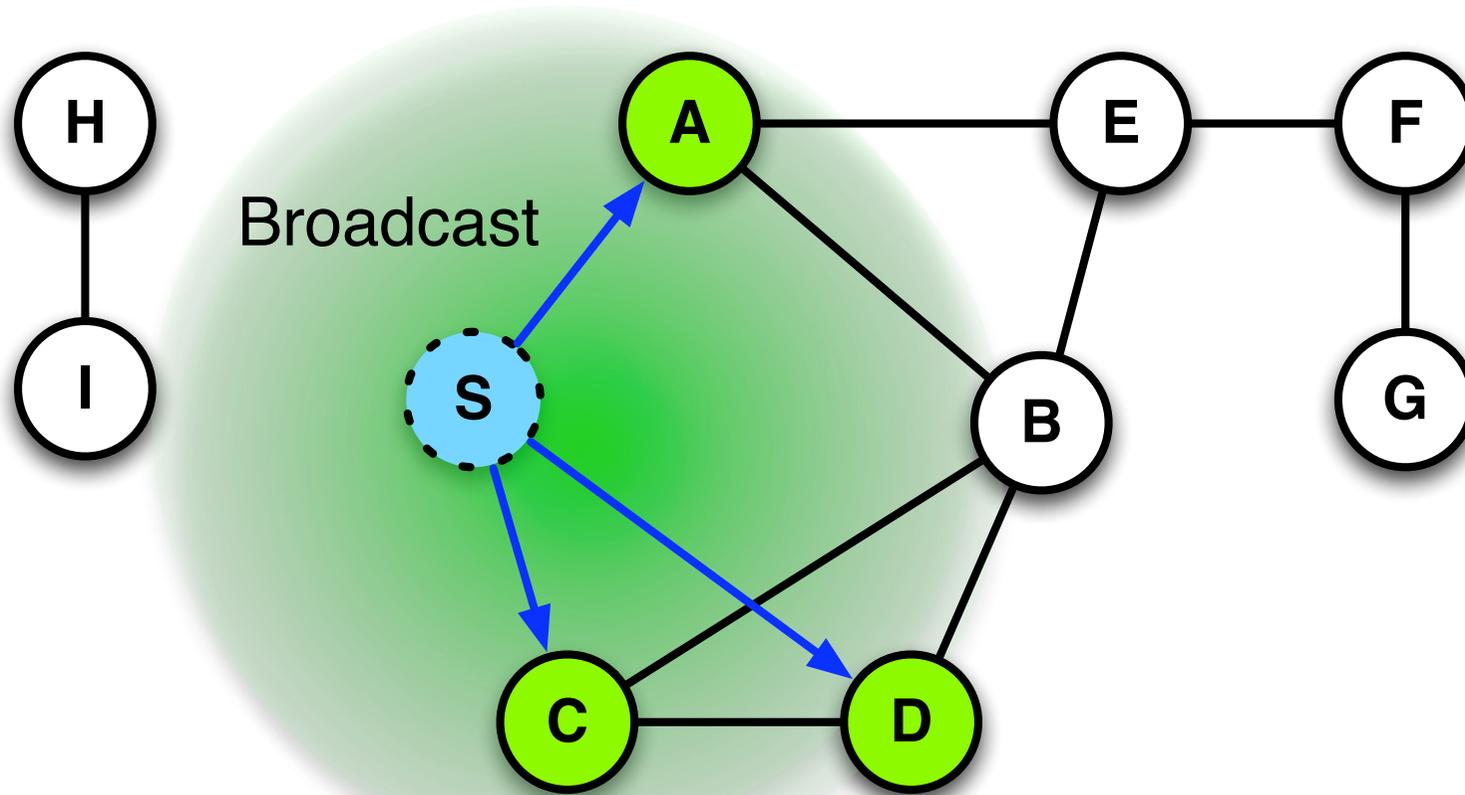
▶ **Paket erreicht immer das Ziel**

- falls es erreichbar ist

Flooding Beispiel

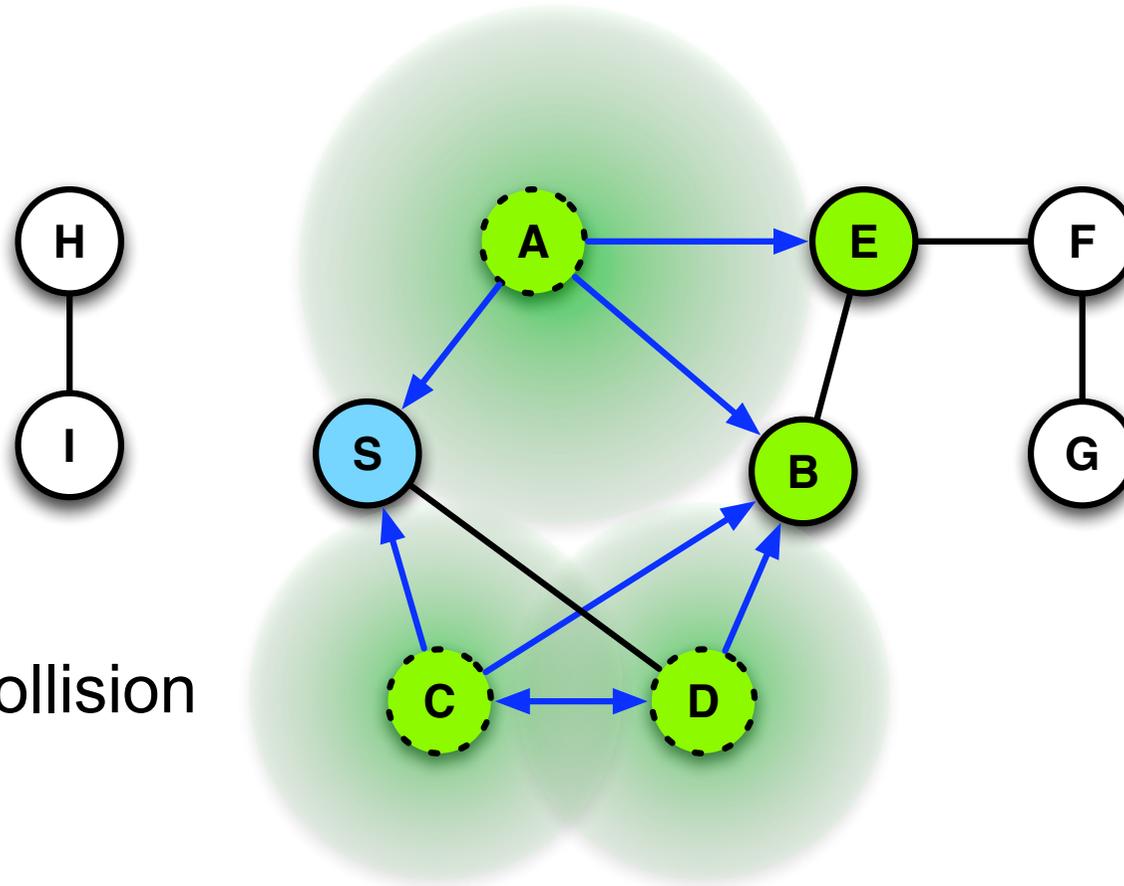


Flooding for Data Delivery



Paket für Empfänger F

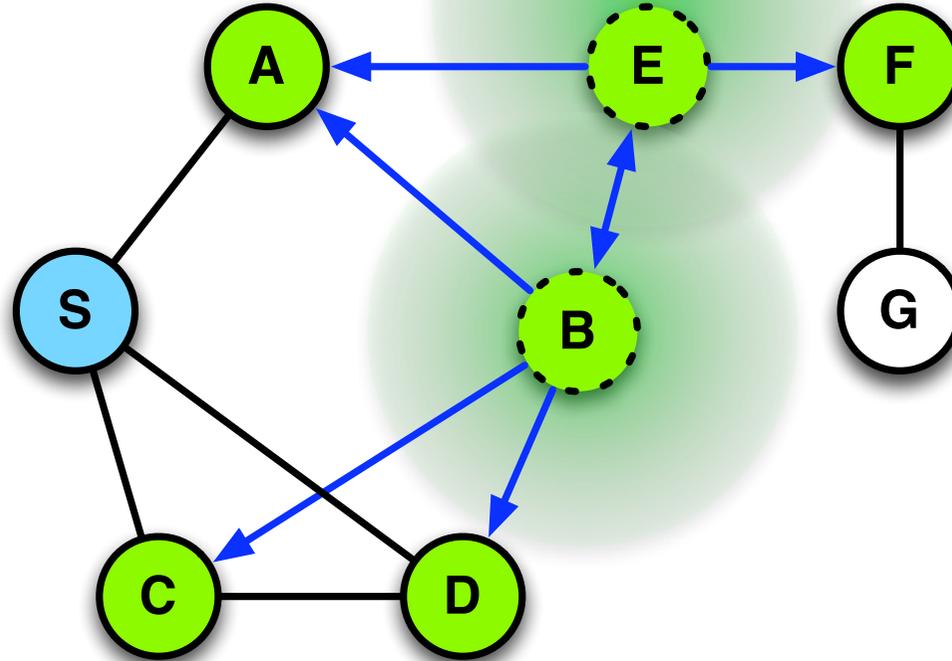
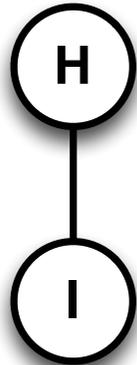
Flooding for Data Delivery



Mögliche Kollision
bei B!

Flooding for Data Delivery

Empfänger F
erhält Paket und
behält es



Knoten G, H, I
erhalten das Paket
nicht

Flooding

▶ Vorteil

- einfach und robust
- für kurze Paketlängen, kleine Teilnehmerzahl in hochmobilen Netzwerken mit wenig Verkehr der beste Ansatz

▶ Nachteil

- Hoher Overhead
- Broadcasting ist unzuverlässig
 - wegen fehlender Bestätigungen
 - hidden, exposed Terminals führen zu Datenverlust oder Verzögerung

Flooding

- ▶ **Erzeugt zu viele unnötige (lange) Datenpakete**
 - im worst case sendet jeder Teilnehmer jedes Paket
 - viele Kollisionen langer Übermittlungen führen zu langen Wartezeiten im Mediumzugriff
- ▶ **Besserer Ansatz:**
 - Verwendung von Kontrollpaketen zur Routenbestimmung
 - Fluten von Kontrollpaket führt zu DSR

Dynamic Source Routing (DSR)

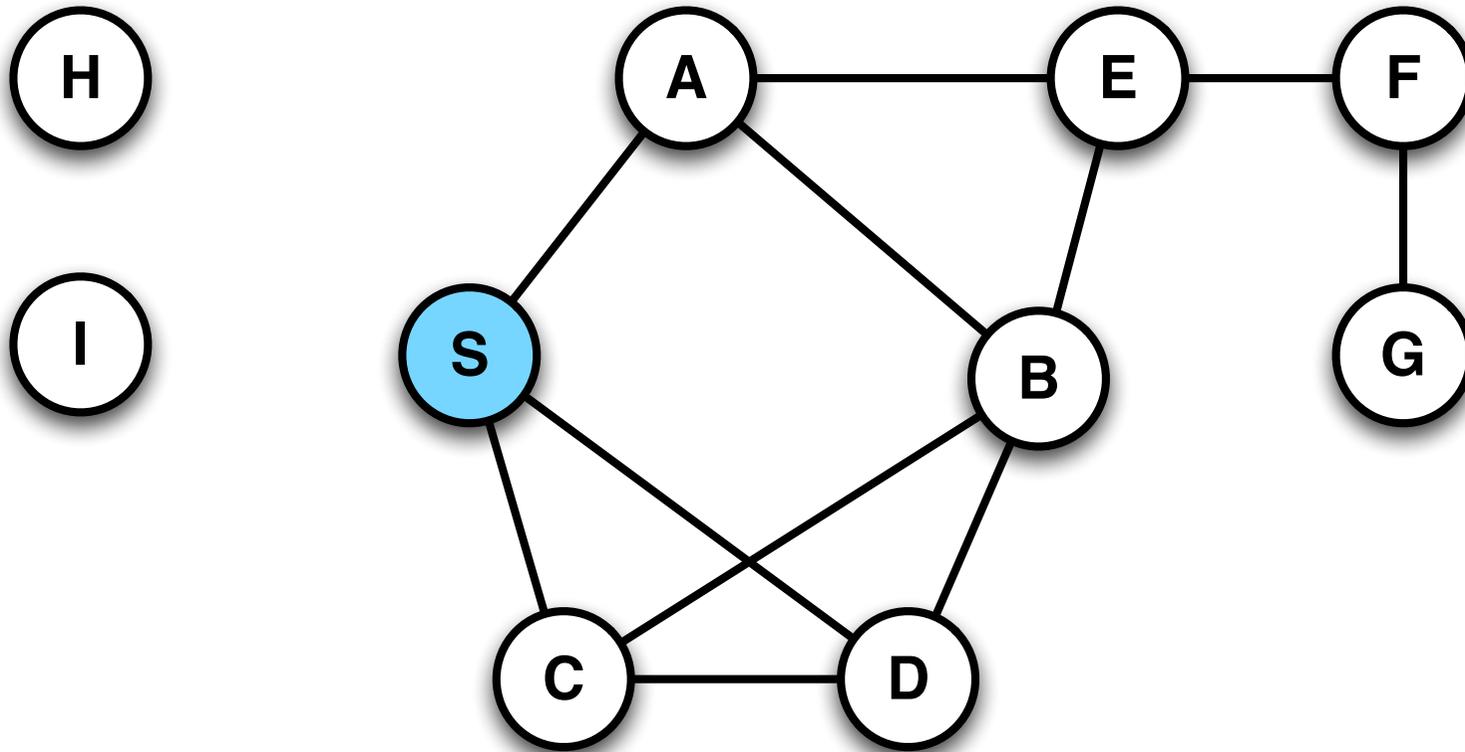
▶ Johnson, Maltz

- *Dynamic Source Routing in Ad Hoc Wireless Networks*, Mobile Computing, 1996

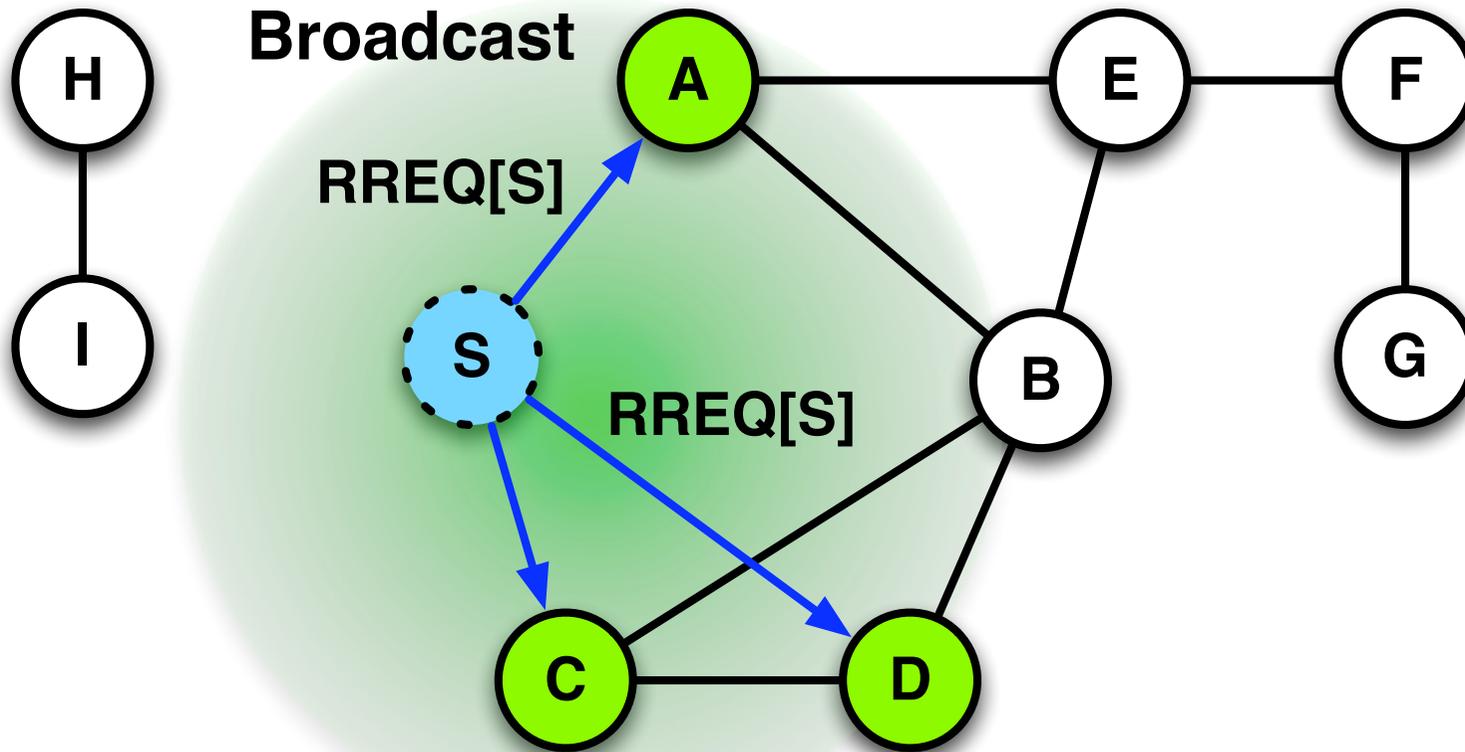
▶ Algorithmus

- Sender initiiert Routen-Ermittlung durch Fluten (Flooding) von **Route-Request (RREQ)**-Paketen
 - Jeder weiterleitende Knoten hängt seine ID an das RREQ-Paket
- Der Empfänger erzeugt aus der Routen-Information des RREQ-Pakets ein **Route-Reply (RREP)**-Paket
 - mit Hilfe der Routeninformation des Pakets wird es zum Sender zurück gesendet
- Sender schickt **Datenpaket** zusammen mit Routeninformation zum Empfänger

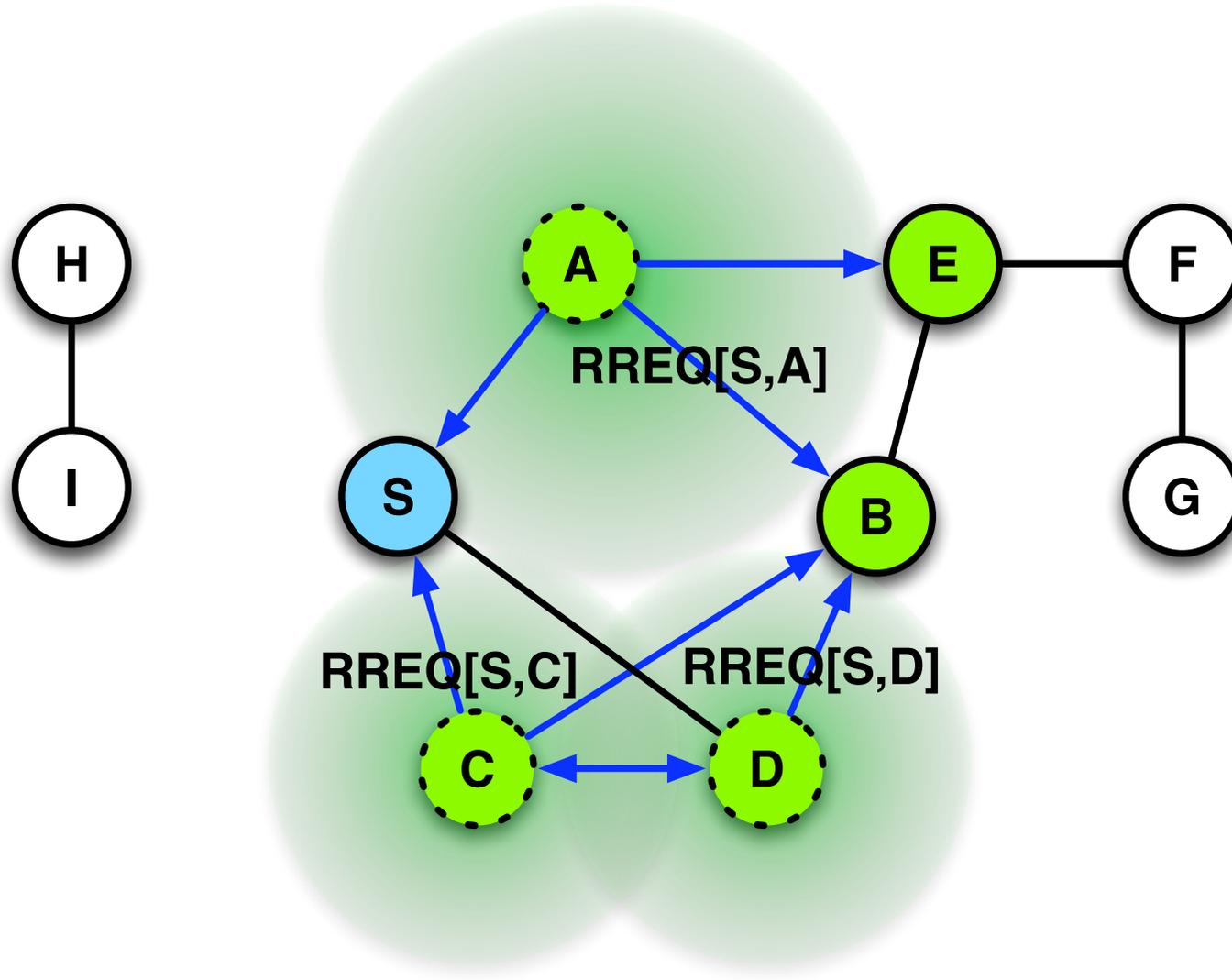
DSR Beispiel



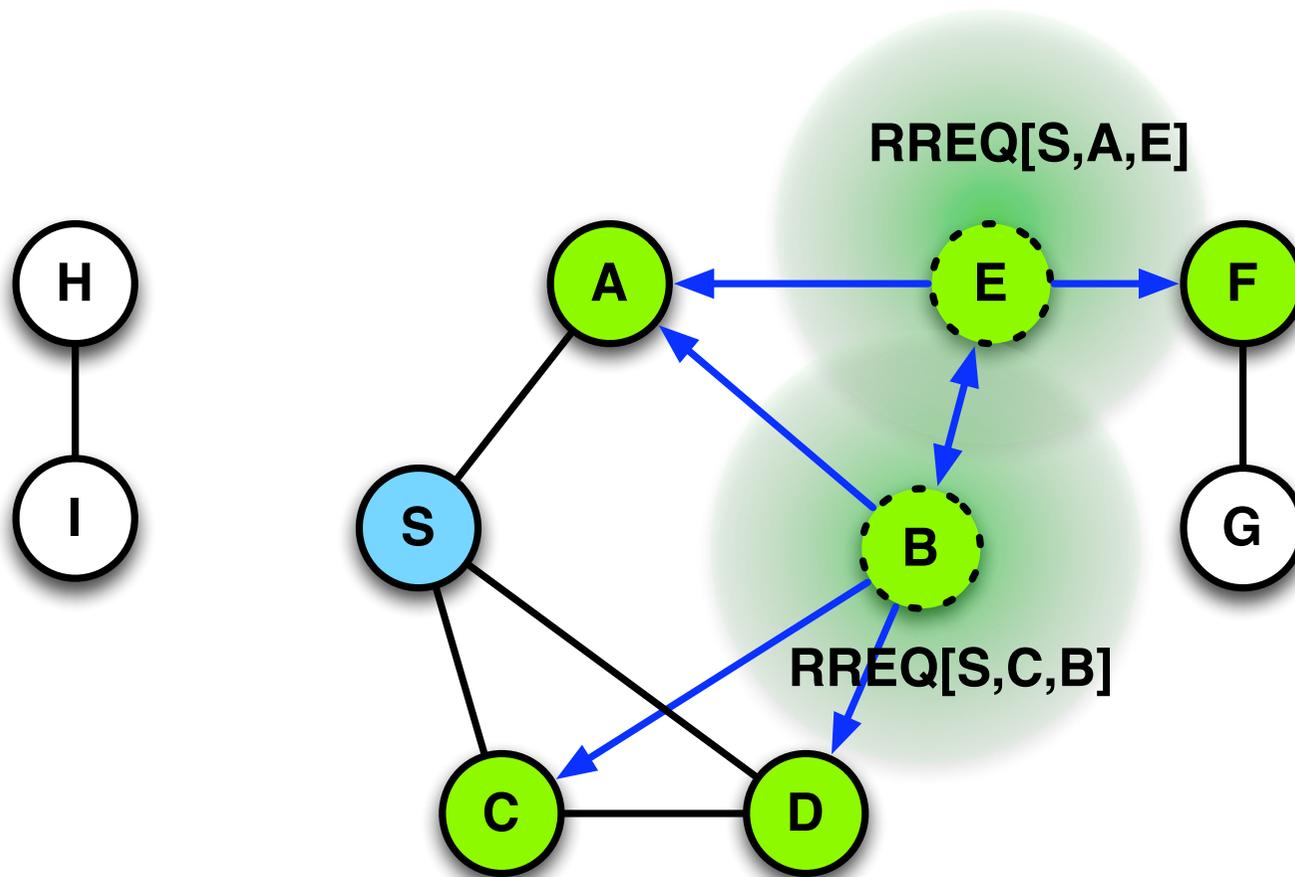
DSR Beispiel



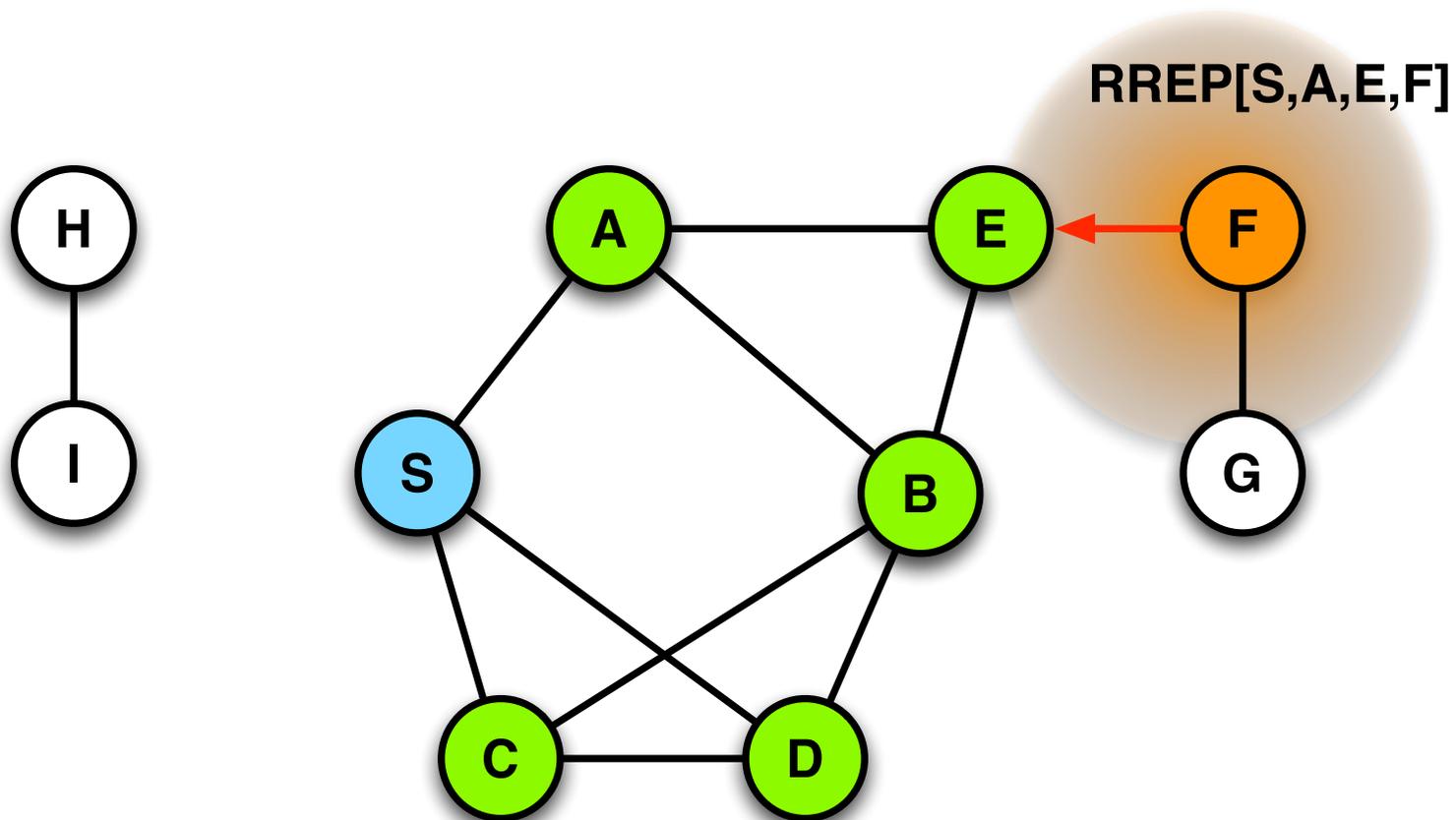
DSR Beispiel



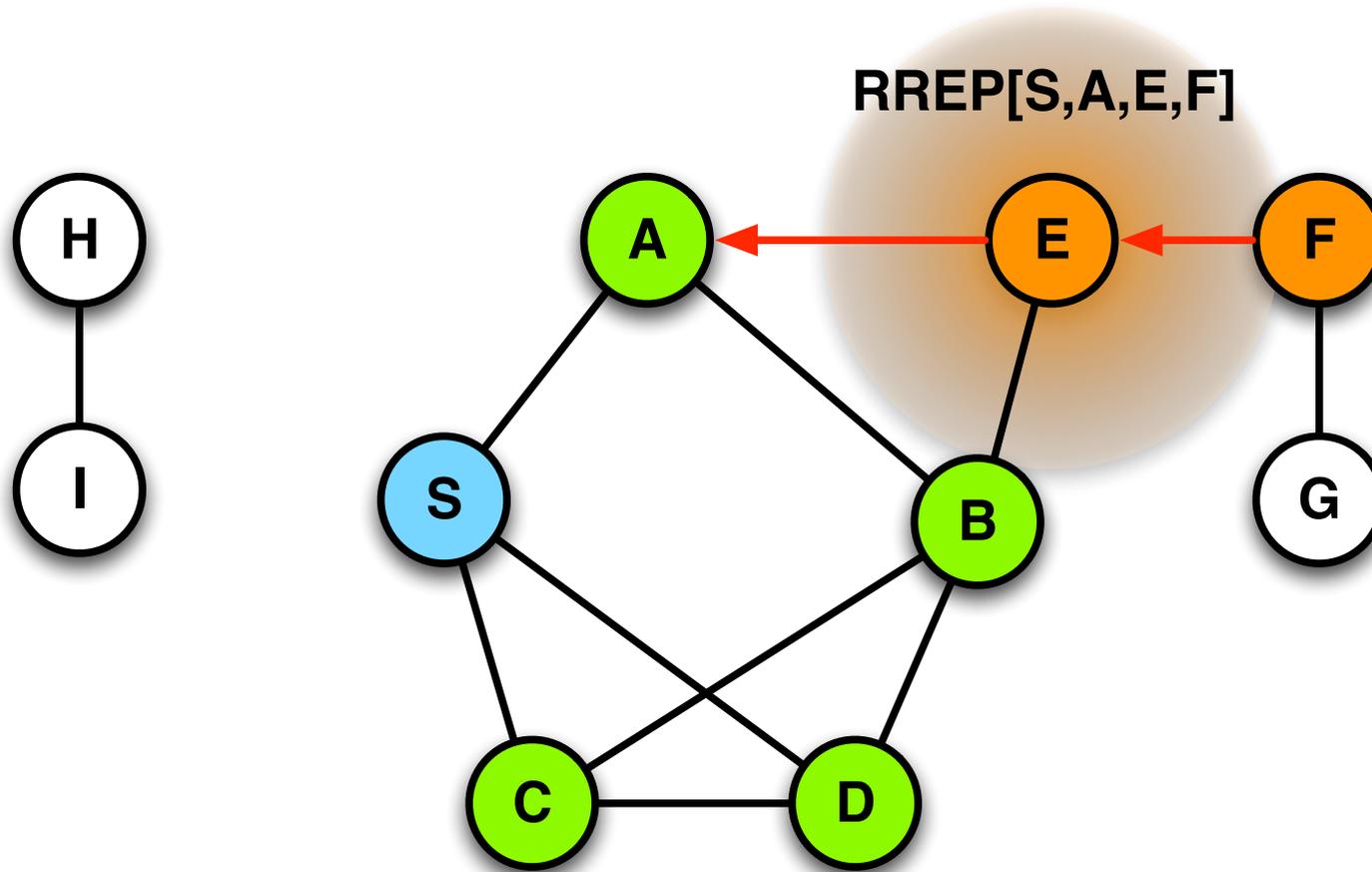
DSR Beispiel



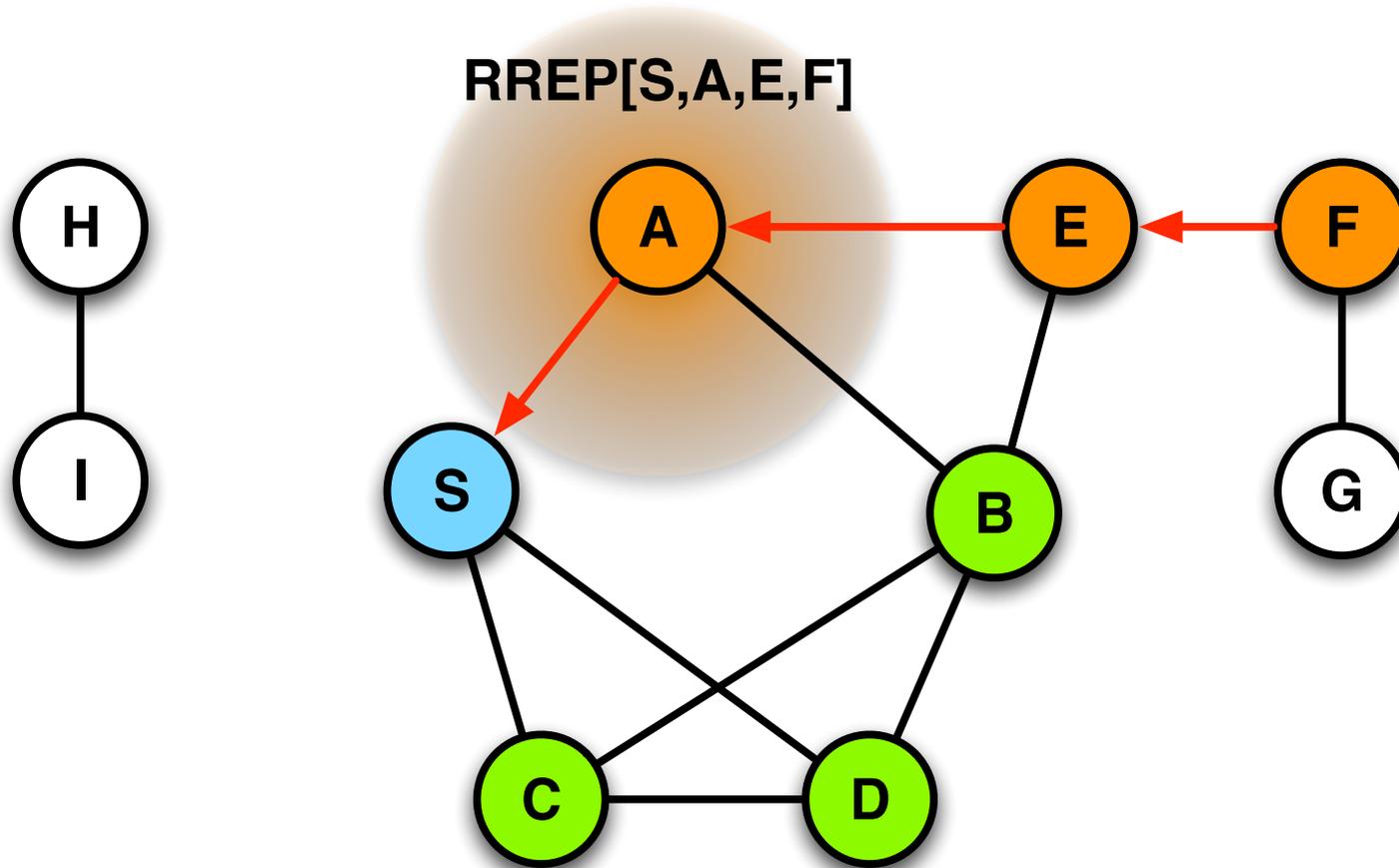
DSR Beispiel



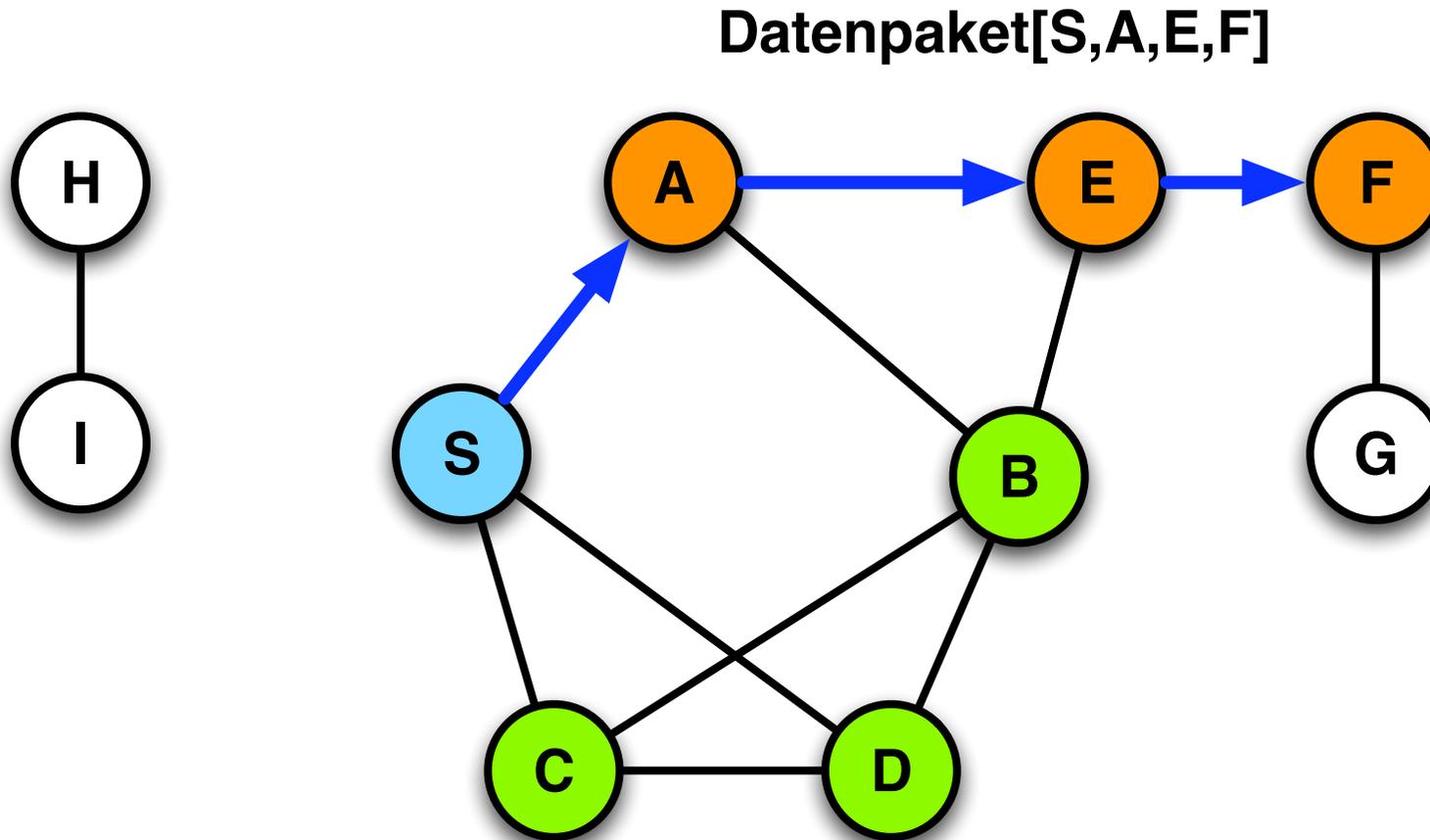
DSR Beispiel



DSR Beispiel



DSR Beispiel



Voraussetzungen

- ▶ **Route Reply**
 - benötigt bidirektionale Verbindungen
 - unidirektionale Verbindungen
 - müssen auf Symmetrie getestet werden
 - oder Route-Reply muss eigenen Route-Request auslösen
- ▶ **Datenpaket hat gesamte Routeninformation im Header**
 - daher: Source-Routing
- ▶ **Routenbestimmung**
 - wenn keine gültige Route bekannt ist

DSR Erweiterungen und Modifikationen

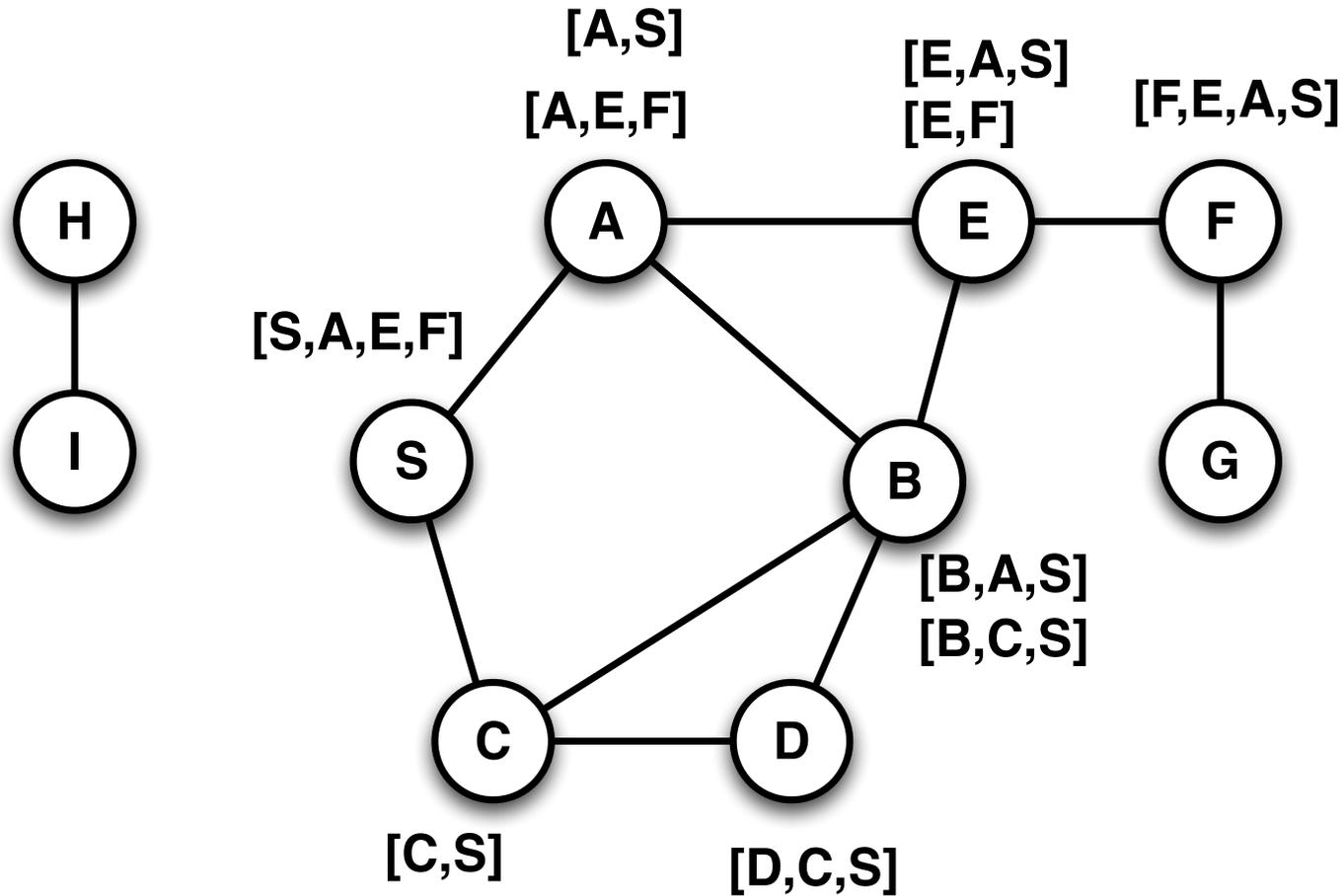
- ▶ **Zwischenrouten können mit Cache-Information RREP generieren**
 - Problem: veraltete Information
- ▶ **Abhören von Kontrollnachrichten**
 - kann helfen die Topologie zu erkennen
- ▶ **Zufällige Verzögerungen für Antworten**
 - Zur Verhinderung von Massen von RREP (Reply-Storm)
 - falls viele Knoten die Antwort kennen (nicht für Mediumszugriff)
- ▶ **Reparatur**
 - Wenn ein Fehler erkannt wird: Üblich Neuberechnung der Route
 - Stattdessen: lokaler Änderung der Source-Route
- ▶ **Cache Management**
 - Mechanismen zur Löschung veralteter Cache-Information

DSR Optimierung

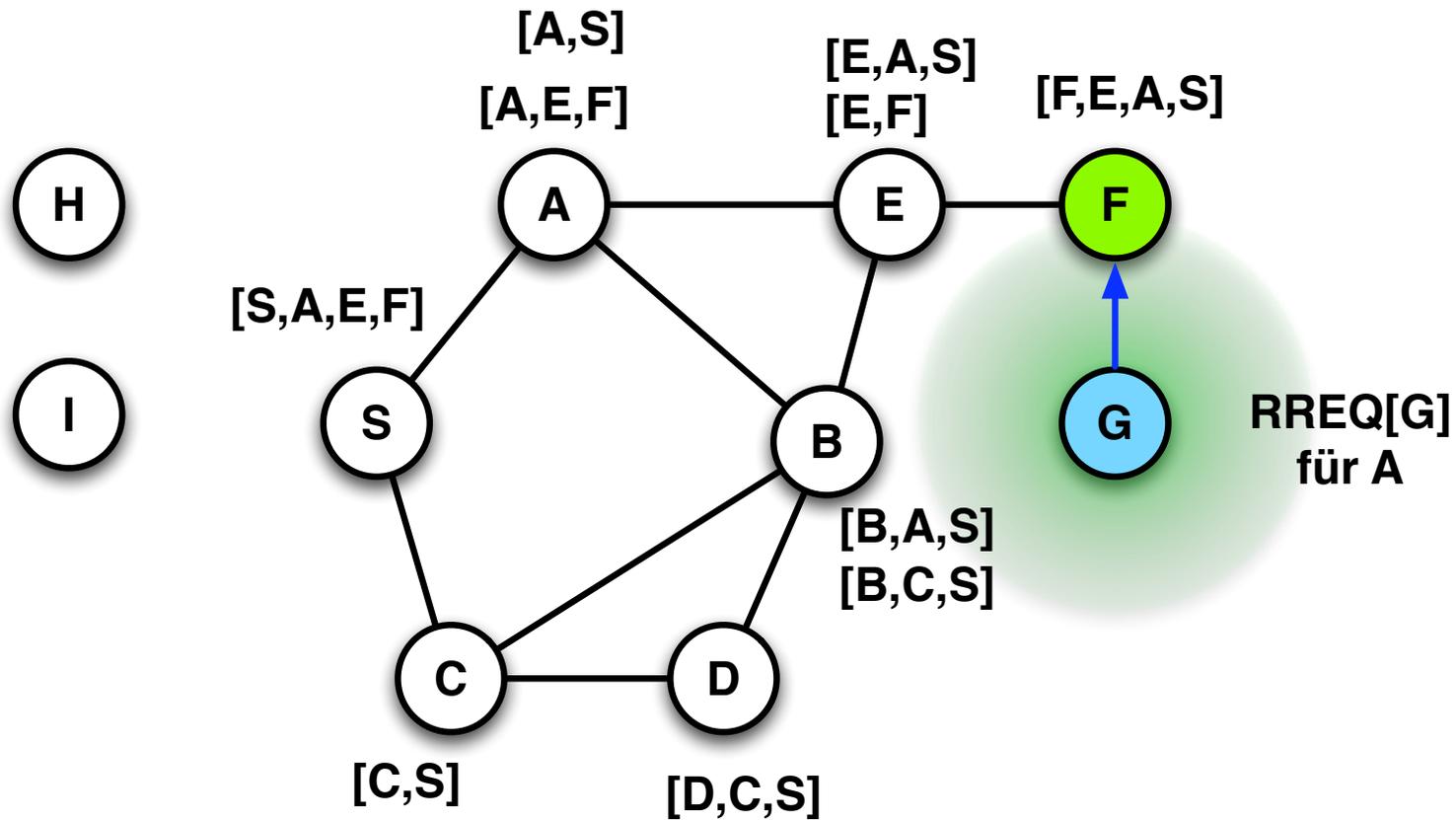
Route Caching

- ▶ **Jeder Knoten speichert alle erhältliche Information aus**
 - Header von Datenpaketen
 - Route Request
 - Route-Reply
 - Teilpfaden
- ▶ **Damit kann ein Route-Reply generiert werden**

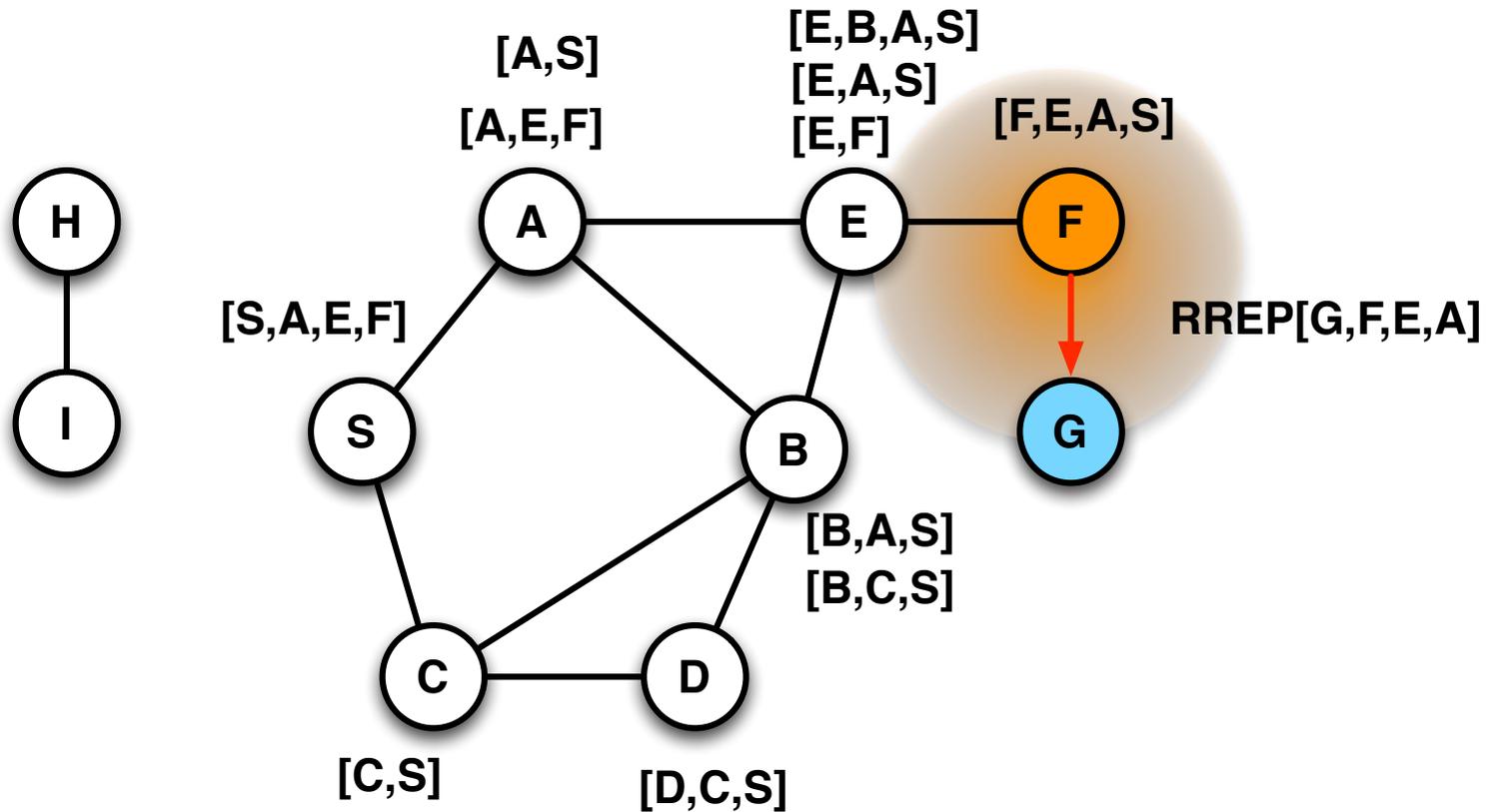
DSR Route Caching



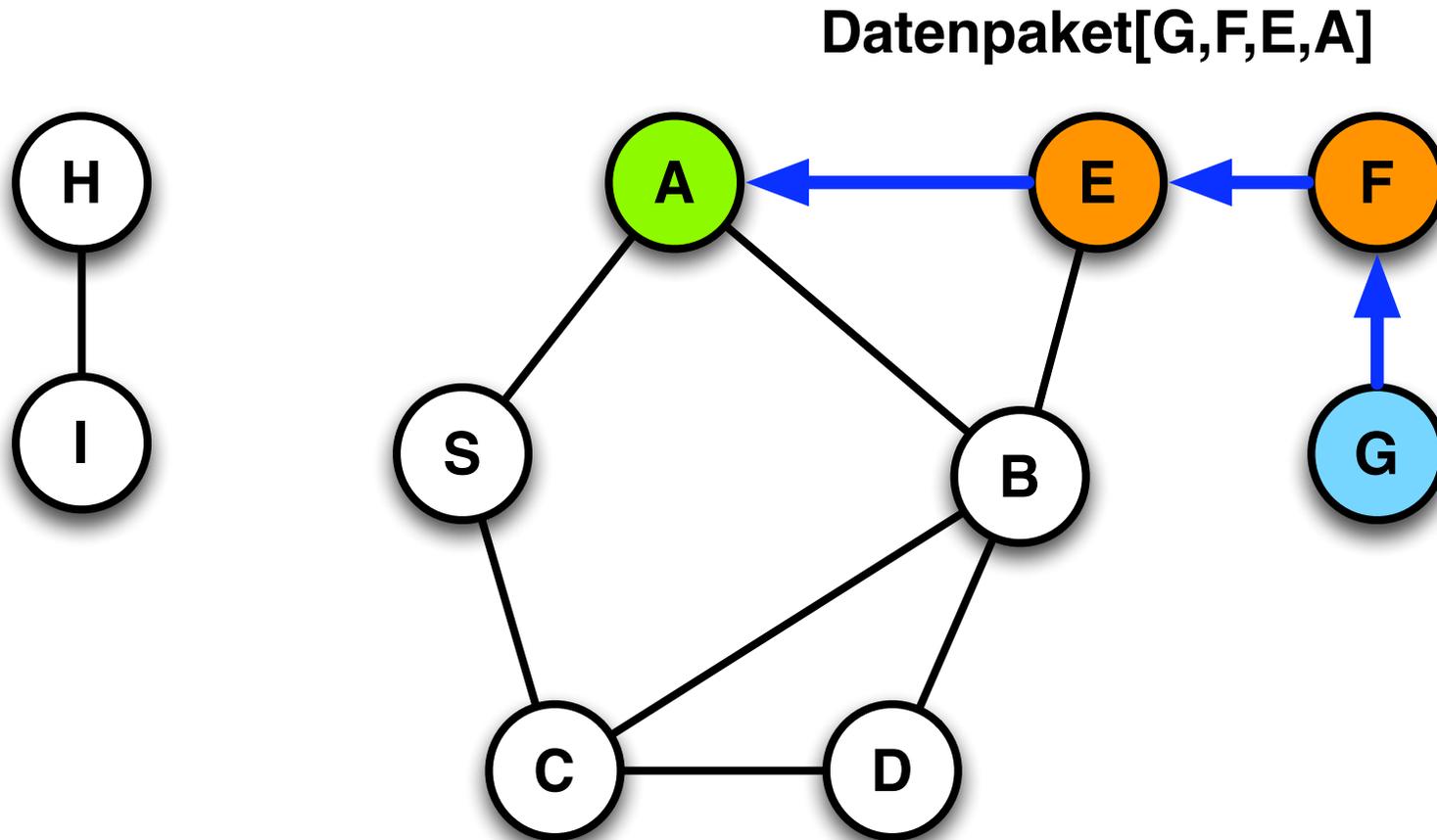
DSR Route Caching



DSR Route Caching



DSR Route Caching

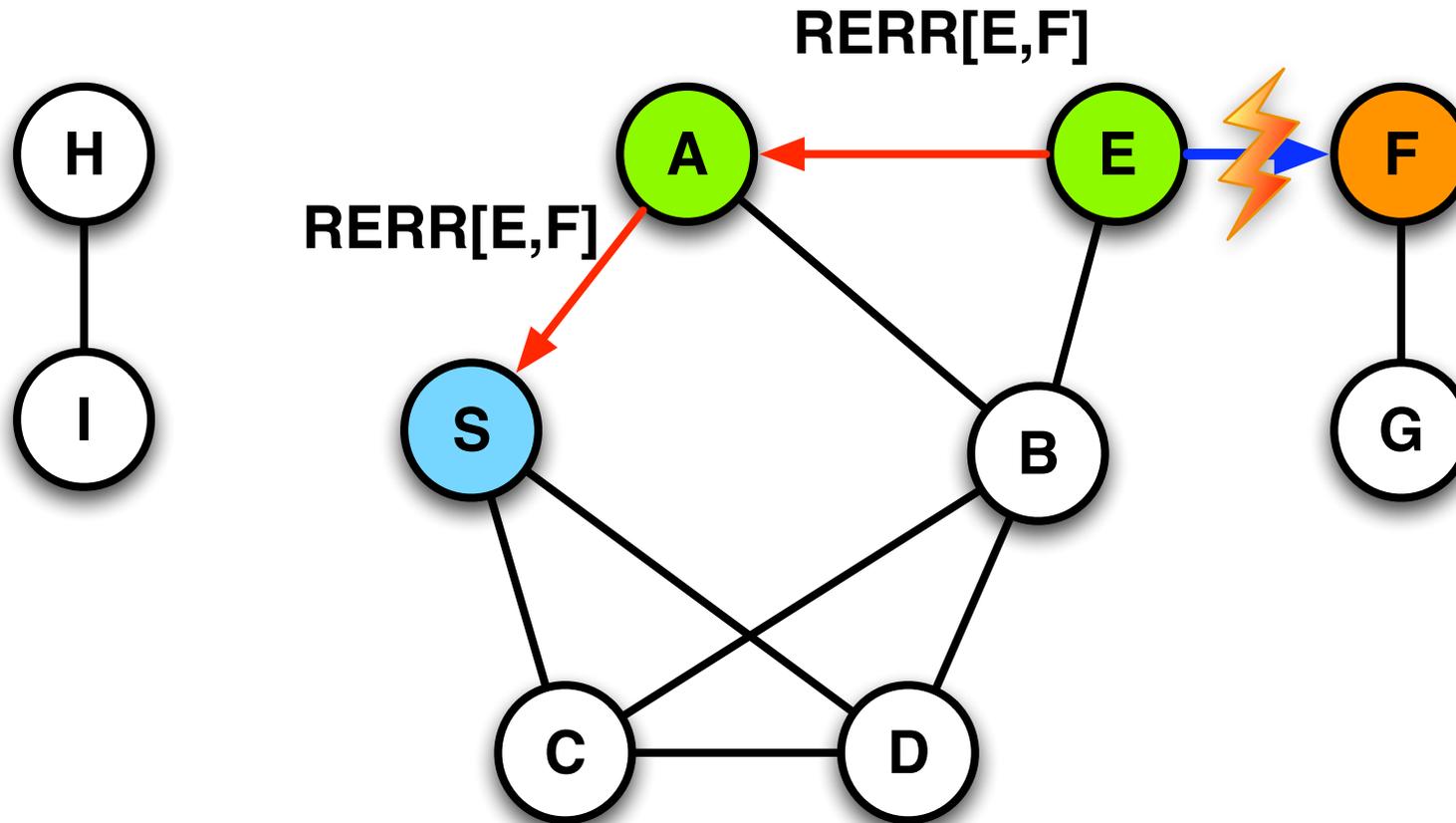


DSR Optimierung

Route Caching

- ▶ **Falls eine Information fehlerhaft ist**
 - weil eine Route nicht mehr existiert
 - dann wird dieser Pfad im Cache gelöscht und
 - alternative Pfade verwendet
 - oder RREQ generiert
- ▶ **Fehlende Links werden durch Route-Error (RERR) Pakete im Netz verbreitet**

Route Error



DSR Diskussion

▶ Vorteile

- Routen werden nur zwischen kommunizierenden Knoten aufrecht erhalten
- Route caching reduziert Routen-Suche
- Caches helfen viele alternative Routen zu finden

▶ Nachteile

- Headergröße wächst mit Entfernung
- Route-Requests flutet möglicherweise Netzwerk
- Route-Reply-Storm
- Veraltete Cache-Information verursachen Overhead



ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Algorithmen für drahtlose Netzwerke

Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Rechnernetze und Telematik
Prof. Dr. Christian Schindelhauer

