

Internet Protokolle II

*Routing in
Wireless Ad-hoc Networks
Part I*

Thomas Fuhrmann

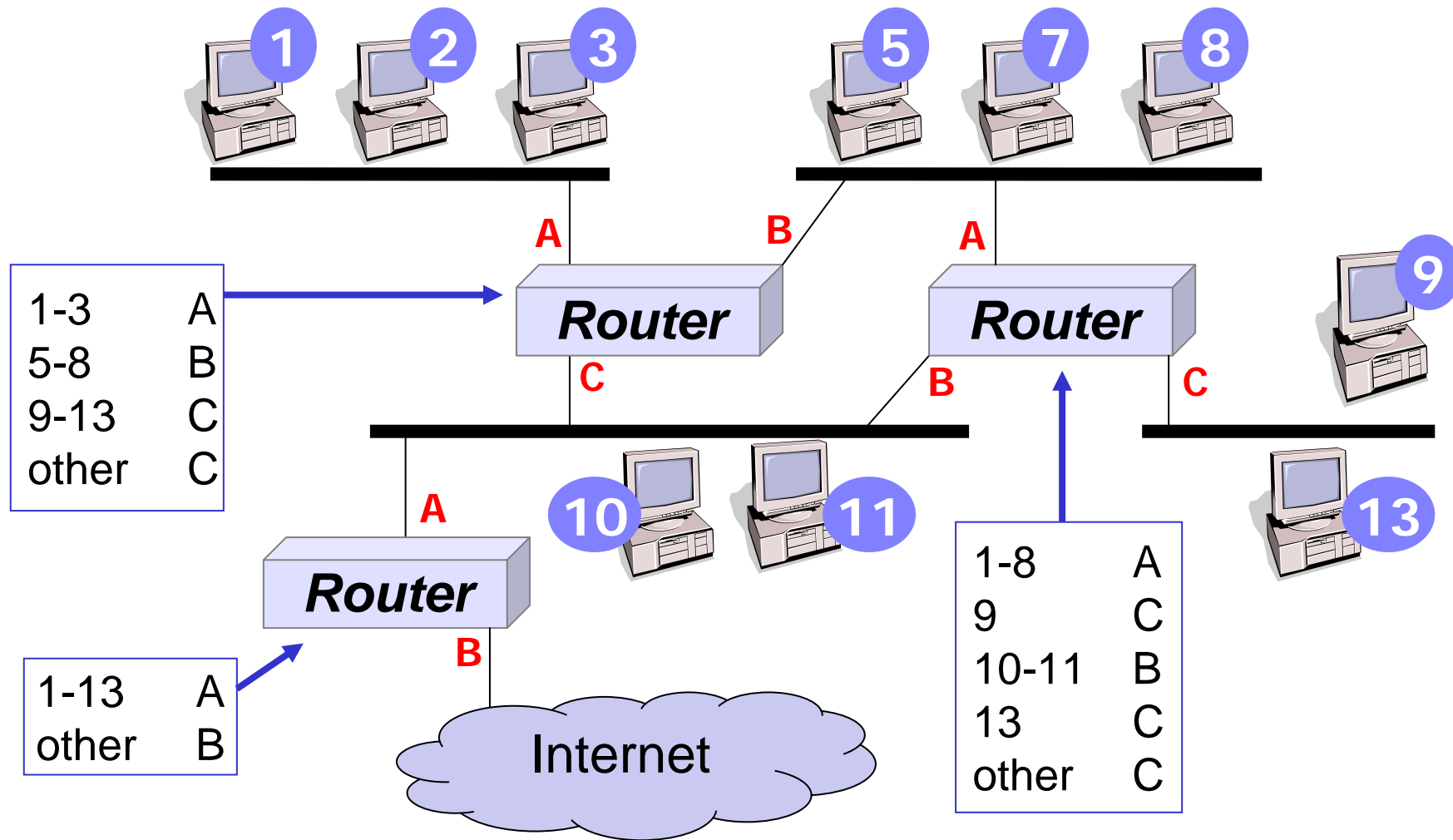


Network Architectures
Computer Science Department
Technical University Munich

Mobile Ad-hoc Networks (MANET)

- Hoch dynamische Netztopologie
 - Durch Mobilität der Geräte und sich verändernde Eigenschaften des drahtlosen Kanals (fading)
 - Auch Partitionierung des Netzes möglich
 - Dadurch Paketverluste
- Asymmetrische Verbindungen
 - Verbindungsqualität kann in beiden Richtungen stark unterschiedlich sein
- Drahtloses Medium ist Broadcast-Medium
 - Versteckte und ausgelieferte Endgeräte
- Begrenzte Batterieleistung der mobilen Geräte
 - Verstärkt durch Signalisierungsverkehr z.B. des Routingprotokolls
- Begrenzte Bandbreite
 - Verstärkt durch Signalisierungsverkehr z.B. des Routingprotokolls und MAC-Protokoll (Kollisionen, versteckte Endgeräte etc.)
- Zeitliche Synchronisation der Geräte schwierig
 - Erschwert z.B. Energiesparmodi der Geräte (z.B. periodisches Schlafen)
- Sicherheitsmechanismen schwierig anzuwenden
 - Abhören des drahtlosen Kanals
 - Jedes Gerät muss Pakete an andere weiterleiten können

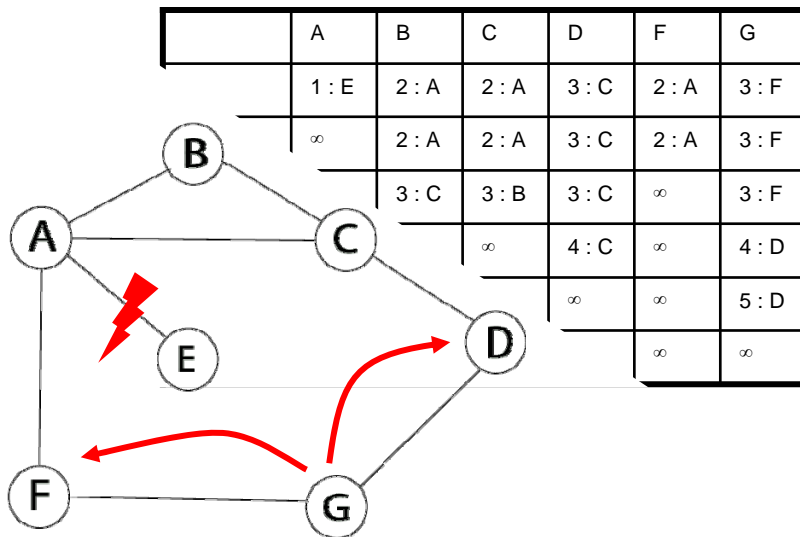
Routing – Recap from Last Term (1)



Routing – Recap from Last Term (2)

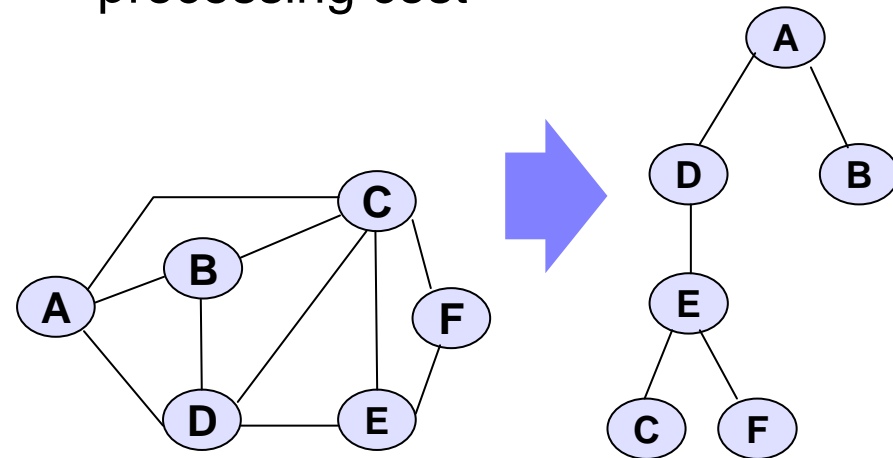
Distance Vector

- Routers exchange their routing tables
- No additional state
- Slow adaptation to failed links

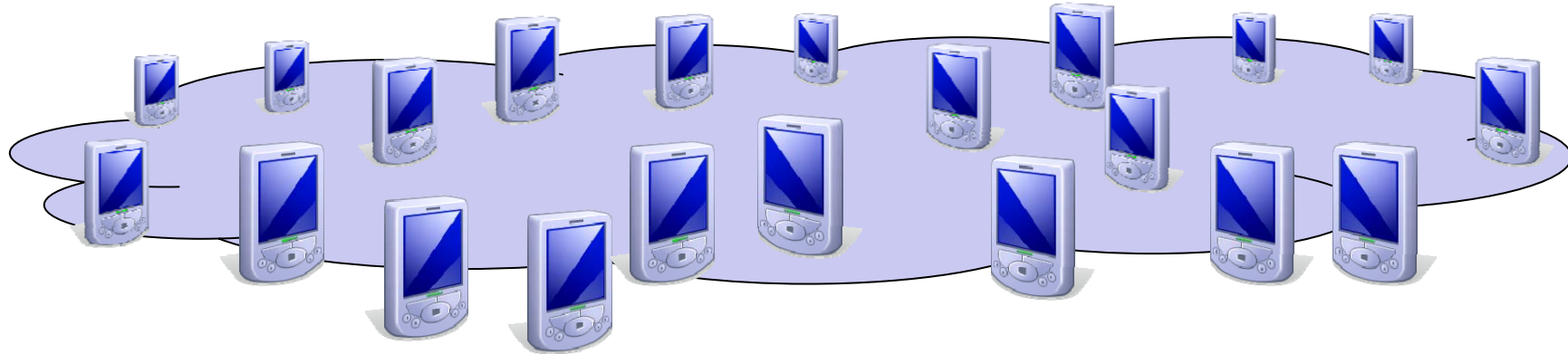


Link State

- Routers flood and store full topology
- Dijkstra to calculate routing table
- Responds flexibly to topology changes
- High memory, bandwidth, and processing cost



Problems with Mobile Ad-hoc Networks



- Nodes move → No easy address aggregation!
 - Mobile IP like approach: But where do you fix the home agent?
 - DynDNS like approach: But where do you fix the look-up service?
- All nodes move
 - Distance vector routing would not converge.
 - Link state routing would cause giant communication and processing overhead.

Proactive and Reactive Routing

- Classic Internet Routing is proactive
 - Routing protocol builds up routing table in case a packet needs to be routed
 - Reasonable in the Internet because
 - large sub-networks and autonomous systems
 - Relatively stable topology
- Mobile Ad-hoc Network Routing could be reactive
 - Wait for a packet; then try to find an appropriate route.
 - Network isn't stable, so why invest effort to build up routing state that won't be used before it is outdated?

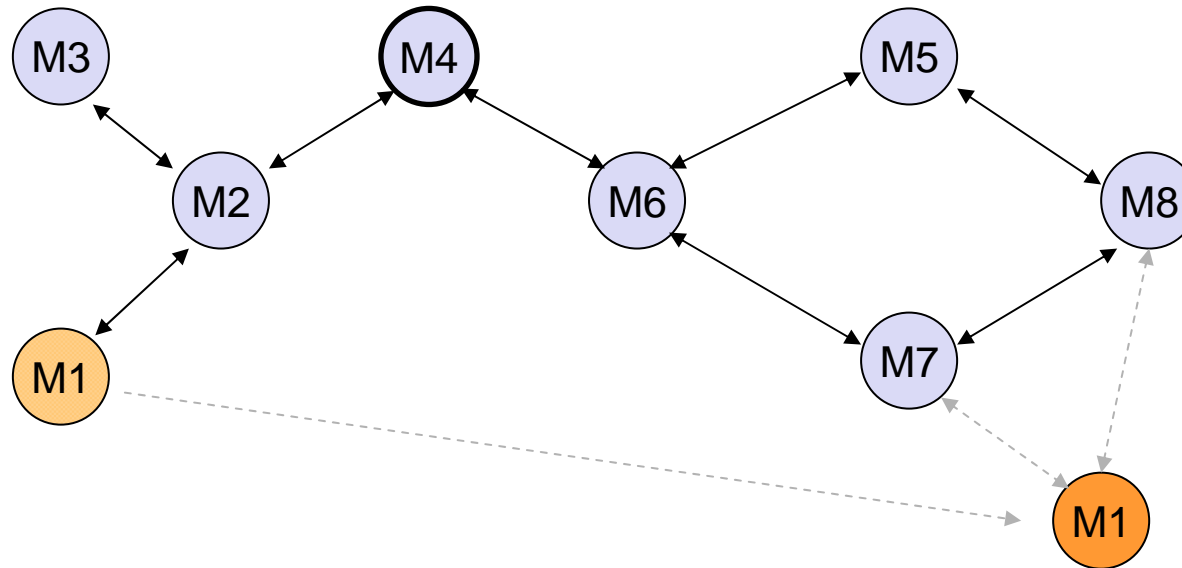
Routing in Mobile Ad-Hoc Networks

- Extension of fixed wire protocols
 - Avoid ‚count to infinity‘ by exchanging richer state info
 - Damp updates to limit traffic
- Flooding to acquire state
 - Do not proactively maintain state
 - Search reactively for the requested destinations
- Limited flooding – Scopes and landmarks
 - Different areas have more or less detailed routing information
- Geographic routing
 - Exploit the fact that nodes have coordinates
- Virtual structures
 - Establish virtual ‚coordinates‘
 - Distribute routing info according to virtual coordinates

Destination-Sequenced Distance Vector Protocol

- Erweiterungen der Routingeinträge um **Sequenznummern**
 - Sequenznummern werden von jeweiligen Zielknoten vergeben
 - Sicherstellung, dass Aktualisierungen in der richtigen Reihenfolge ausgeführt werden
 - Vermeidet dadurch Schleifen und Inkonsistenzen
- Aufbau / Unterhalten einer Routingtabelle mit
 - Zieladresse, Next Hop, Anzahl der Hops zum Ziel, Sequenznummer, letzte Änderung des Eintrags und Zeiger zu alternative Routen
- Periodisches Versenden der Routinginformation
 - Einträge der versendete Routingtabelle enthalten
 - Teilmenge der eigenen Routingtabelle:
Zieladresse, Anzahl der Hops zum Ziel, Sequenznummer
 - Neue Sequenznummer für eigenen Eintrag
 - Signifikante Änderungen werden sofort verschickt
 - Linkänderungen aufgrund von Mobilität
 - Erkennbar durch Hello-Nachrichten oder MAC-Acknowledgements
 - „Toter Link“ wird mit Metrik ∞ versehen
 - Unterscheidung zwischen
 - *full dump*: Enthält Einträge aller Knoten
 - *incremental dump*: Enthält nur geänderte Einträge
- DSDV berücksichtigt nur bidirektionale Links

Beispiel DSDV



- Ad-hoc Netz mit mobilem Knoten M1
 - M1 bewegt sich von M2 zur Nachbarschaft von M7 und M8
 - Im folgenden wird der mobile Knoten M4 und dessen Routingtabelle genauer betrachtet

Bsp: Routingtabelle von M4 vor dem Wechsel

... bevor M1 in die Nachbarschaft von M7 und M8 „gewandert“ ist

Ziel	Next Hop	Metrik	Seq.-Nr.	Install	Stable_Data
M1	M2	2	S406_M1	T001_M4	Ptr1_M1
M2	M2	1	S128_M2	T001_M4	Ptr1_M2
M3	M2	2	S564_M3	T001_M4	Ptr1_M3
M4	M4	0	S710_M4	T001_M4	Ptr1_M4
M5	M6	2	S392_M5	T002_M4	Ptr1_M5
M6	M6	1	S076_M6	T001_M4	Ptr1_M6
M7	M6	2	S128_M7	T002_M4	Ptr1_M7
M8	M6	3	S050_M8	T002_M4	Ptr1_M8

- *Install*-Zeit aller Einträge sind etwa gleich
 - Alle mobilen Knoten waren M4 ungefähr zum gleichen Zeitpunkt bekannt
- *Stable_Data* sind Zeiger auf eine Liste alternativer Routen
 - *Ptr1_Mi* sind alles Nullpointer, da jeweils nur ein möglicher nächster Nachbar (*Next Hop*) pro Zielknoten existiert

Bsp: Routingtabelle von M4 nach dem Wechsel

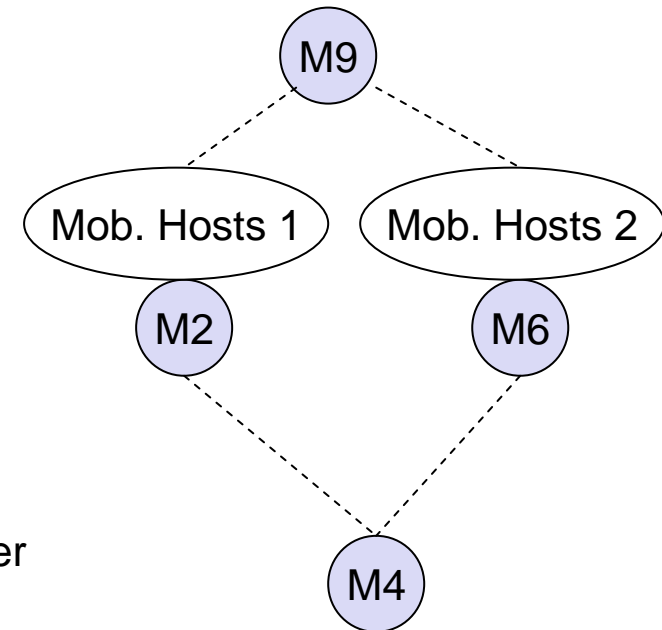
- ... nachdem M1 in die Nachbarschaft von M7 und M8 „gewandert“ ist

Ziel	Next Hop	Metrik	Seq.-Nr.	Install	Stable_Data
M1	M6	3	S516_M1	T810_M4	Ptr1_M1
M2	M2	1	S128_M2	T001_M4	Ptr1_M2
M3	M2	2	S564_M3	T001_M4	Ptr1_M3
M4	M4	0	S710_M4	T001_M4	Ptr1_M4
M5	M6	2	S392_M5	T002_M4	Ptr1_M5
M6	M6	1	S076_M6	T001_M4	Ptr1_M6
M7	M6	2	S128_M7	T002_M4	Ptr1_M7
M8	M6	3	S050_M8	T002_M4	Ptr1_M8

- Die Mobilität von M1 veranlasst
 - Senden eines Incremental Dump von M1
 - Dieses gelangt über M7, M6 zu M4
 - M4 aktualisiert seine Routingtabelle

Dämpfung von Änderungen

- Ziel
 - Vermeidung von Fluktuationen in den Routen
- Änderungen werden entsprechend folgender Kriterien berücksichtigt
 - Routen mit neueren Sequenznummern werden bevorzugt
 - Bei gleichen Sequenznummern werden Routen mit der besseren Metrik bevorzugt
- Szenario
 - Alle mobilen Knoten übertragen Update ungefähr alle 15 Sekunden
 - M2 hat Route zu M9 mit 12 Hops
 - M6 hat Route zu M9 mit 11 Hops
 - Routinginfo von M2 trifft bei M4 ca. 10s vor M6 ein
 - Daher oszilliert der nächste Hop von M4 zu M9 zwischen M2 und M6
- Lösung
 - Speichern der Zeitdauer zwischen erster und bester Ankündigung eines Weges
 - Zurückhalten einer Aktualisierung, wenn sie vermutlich nicht stabil ist (basierend auf der gespeicherten Zeit)

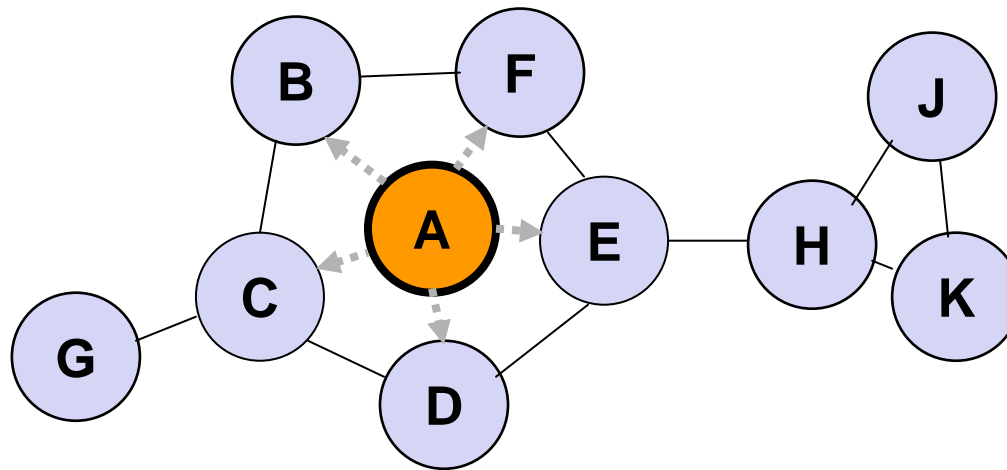


Optimized Link State Routing (OLSR)

- Minimierung des Overheads dadurch, dass Link Status Informationen nur von bestimmten Knoten weitergeleitet werden
- Jedem Knoten wird mindestens ein **Multipoint Relay** zugeordnet
- Multipoint Relays sind bestimmte Nachbarn von X, so dass jeder 2-hop Nachbar von X ein 1-hop Nachbar von wenigstens einem Multipoint Relay ist
- Ermittelt werden diese durch Versenden periodischer Nachrichten, die eine Liste der 1-hop Nachbarn des jeweiligen Knotens enthalten

Beispiel OLSR (1)

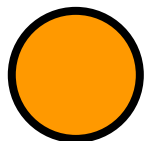
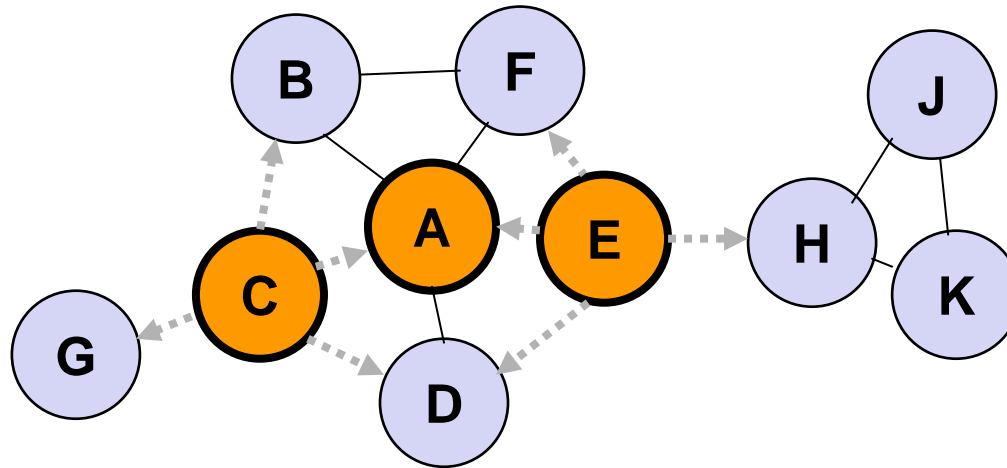
Knoten A versendet Link Status Informationen (Broadcast)



Quelle: N. Vaidya, Tutorial Mobicom 01

Beispiel OLSR (2)

Knoten C und E sind Multipoint Relays von A und leiten daher Link Status Informationen von A weiter

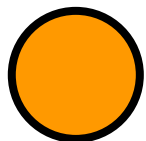
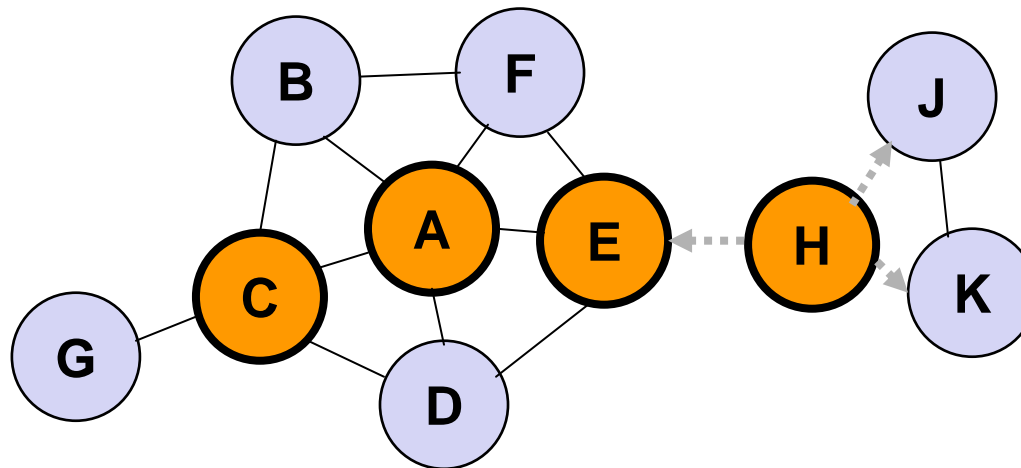


Knoten, der Link Status Informationen von A weiterleitet

Quelle: N. Vaidya, Tutorial Mobicom 01

Beispiel OLSR (3)

Knoten H ist Multipoint Relay von E und leitet daher Link Status Informationen weiter



Knoten, der Link Status Informationen von A weiterleitet

Quelle: N. Vaidya, Tutorial Mobicom 01

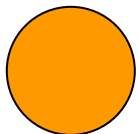
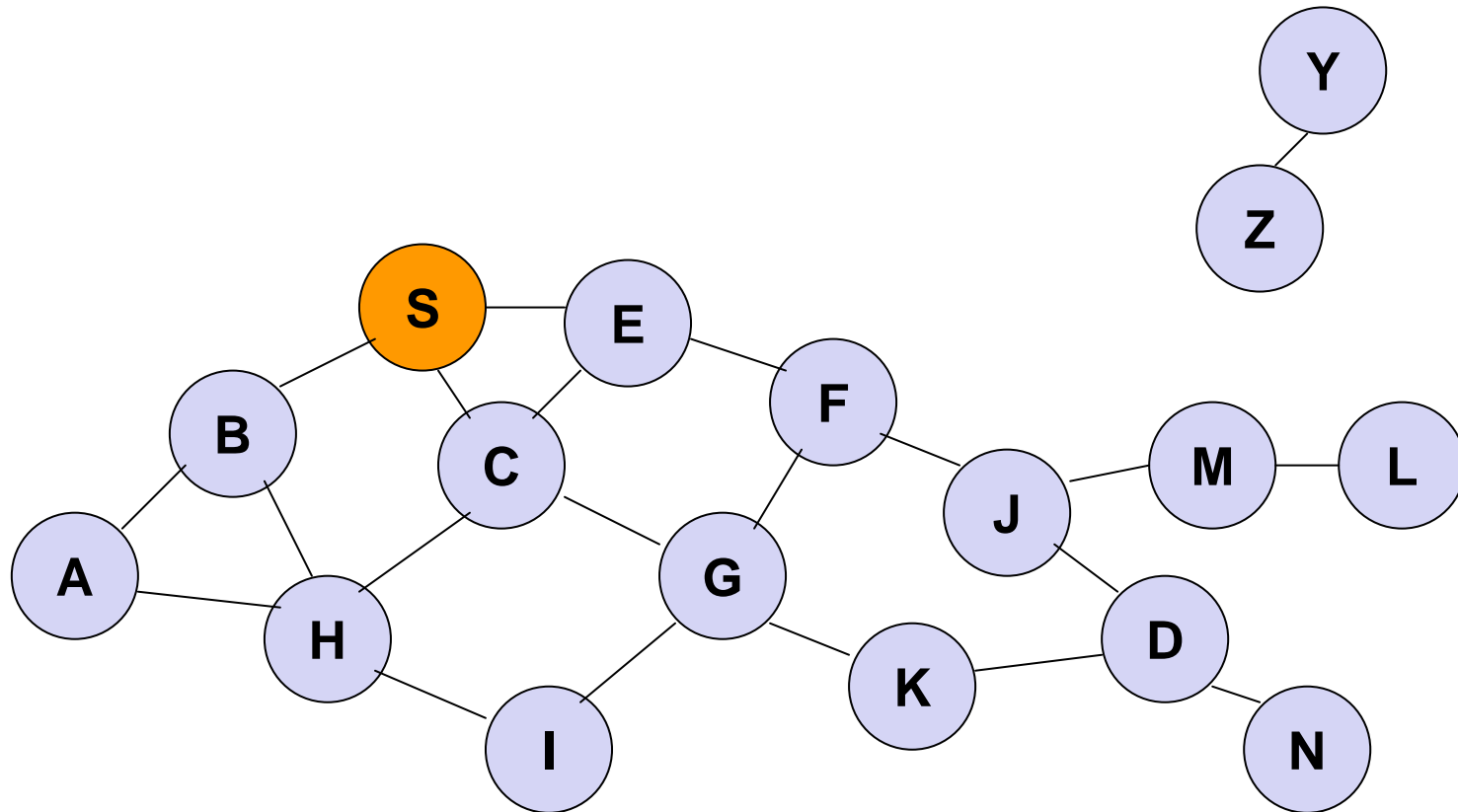
Dynamic Source Routing (DSR)

- Annahmen
 - Durchmesser des Netzes ist meist klein
 - Mobilität der Knoten ist nicht zu hoch
 - Knoten stellen ggfs. „Promiscuous“-Modus zur Verfügung
 - Unidirektionale Übertragungsabschnitte evtl. vorhanden
- Trennung der Routing-Aufgabe in Auffinden und Aufrechterhalten
 - **Auffinden** eines Wegs
 - Nur wenn wirklich ein Weg zum Senden von Daten zu einem bestimmten Ziel benötigt wird und noch keiner vorhanden ist
 - **Aufrechterhaltung** eines Wegs
 - Nur wenn ein Weg aktuell benutzt wird muss dafür gesorgt werden, dass er weiterhin funktioniert
- Keine periodischen Aktualisierungen!

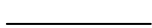
Auffinden eines Pfads bei DSR

- Knoten speichern Pfade (Source Routen) zu den Knoten mit denen sie kommunizieren
- Annahme: Sender hat noch keinen Weg zu einem (neuen) Empfänger im Cache
- Protokoll zum Auffinden des Wegs
 - Sender flutet das Netz mit einem **Route Request-Paket**
 - RREQ enthält Zieladresse und eindeutige ID für dieses RREQ
 - Bei Empfang eines Route Request-Pakets
 - Falls Paket bereits früher erhalten (ID gleich), RREQ verwerfen
 - Endgültiger Empfänger? Dann **Route Reply-Paket** an ursprünglichen Sender schicken
 - Sonst *eigene Adresse anhängen* und per Broadcast weiterleiten
 - Sender erhält Route Reply mit aktuellem Weg (**Source Route**) zurück
 - Nicht unbedingt über den gleichen Weg (bei unidirektionalen Links)
 - Falls MAC unidirektionale Links zulässt, muss Empfänger erst einen Weg zum Sender mit Hilfe eines Route Requests auffinden

Beispiel DSR (1)



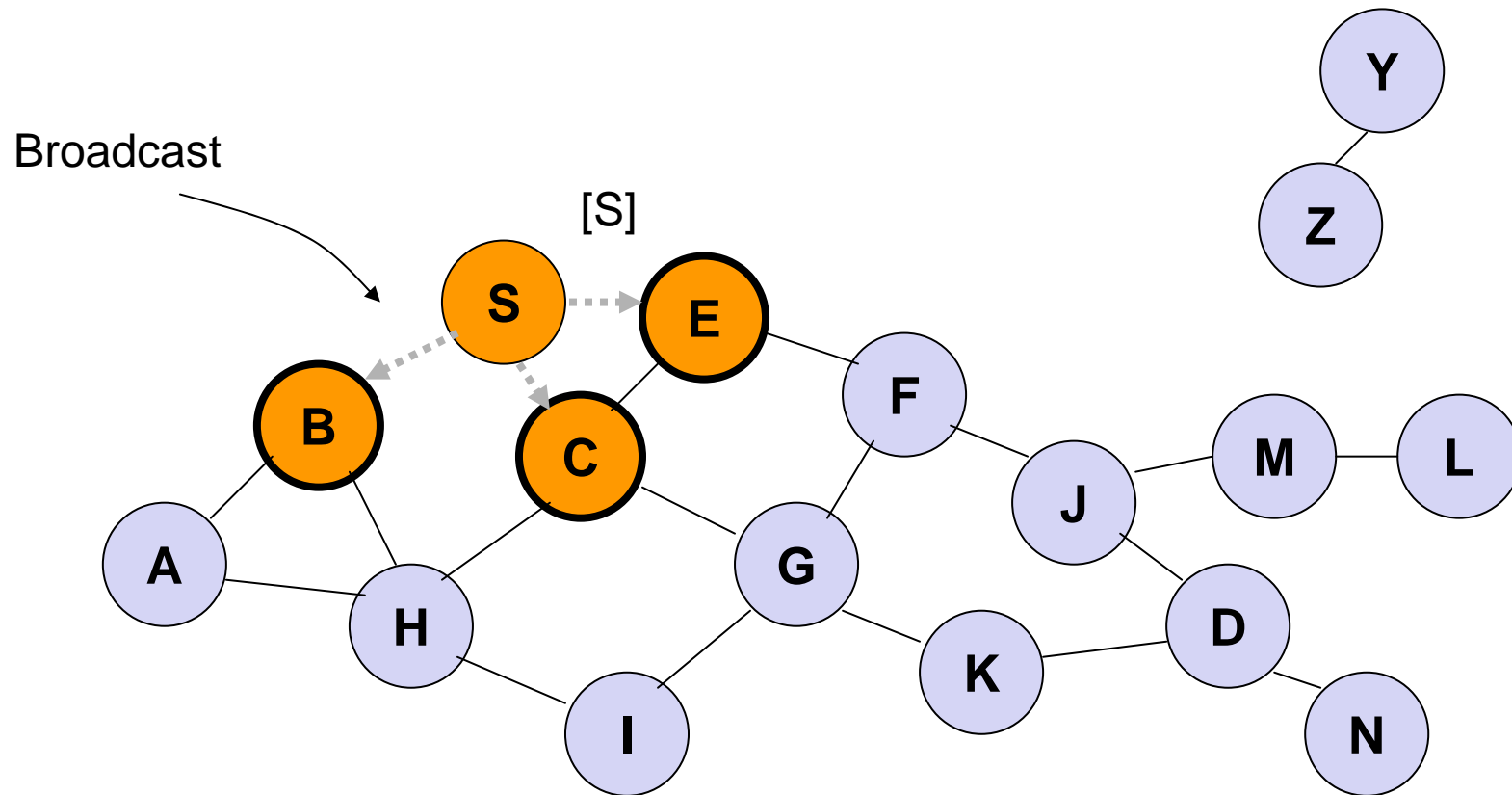
Knoten, der Route Request (RREQ) empfangen hat



Knoten in Sende/Empfangsreichweite

Quelle: N. Vaidya, Tutorial Mobicom 01

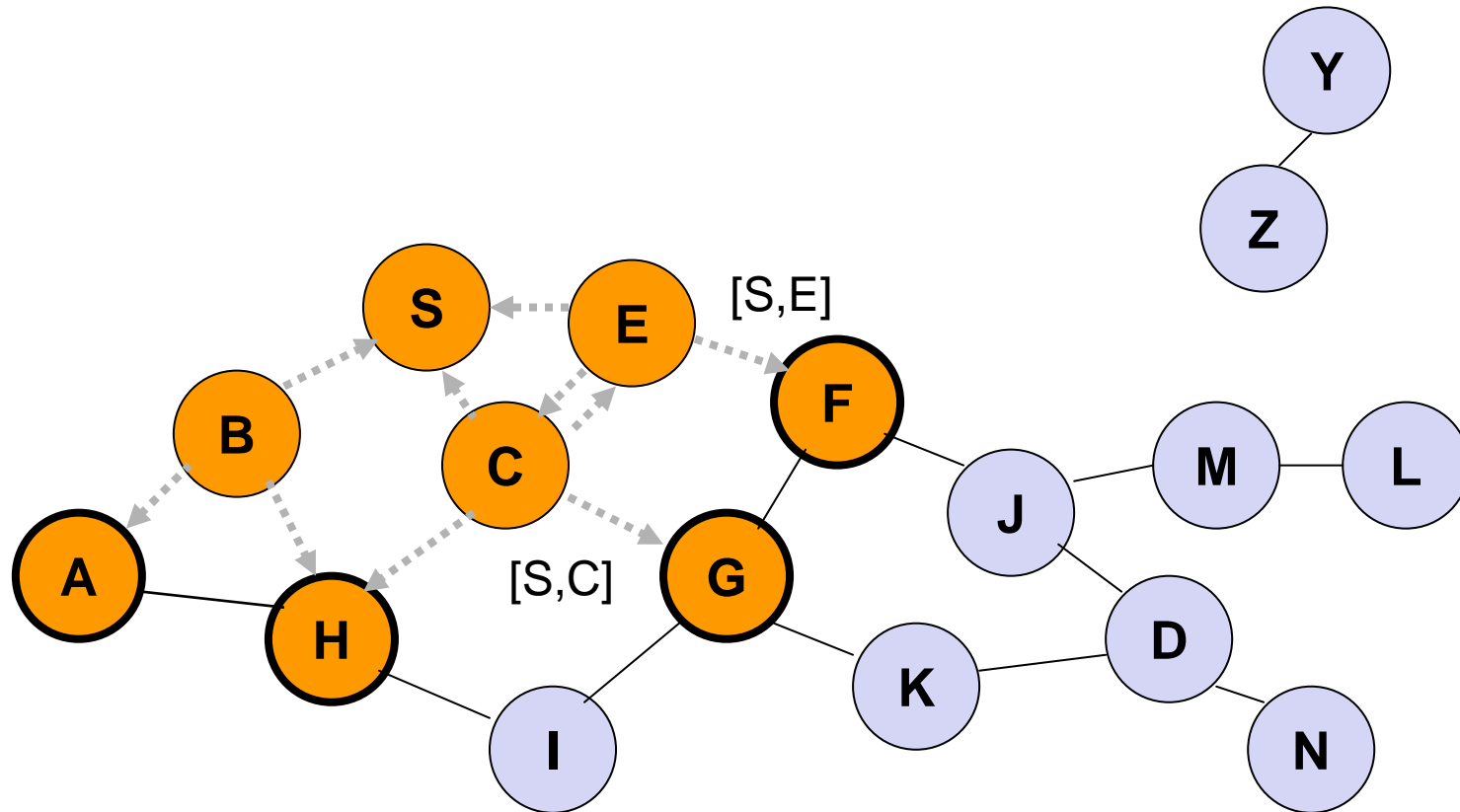
Beispiel DSR (2)



- ❑ Sender S versendet RREQ (Broadcast)
- ❑ Eigene Adresse wird an RREQ angehängt

Quelle: N. Vaidya, Tutorial Mobicom 01

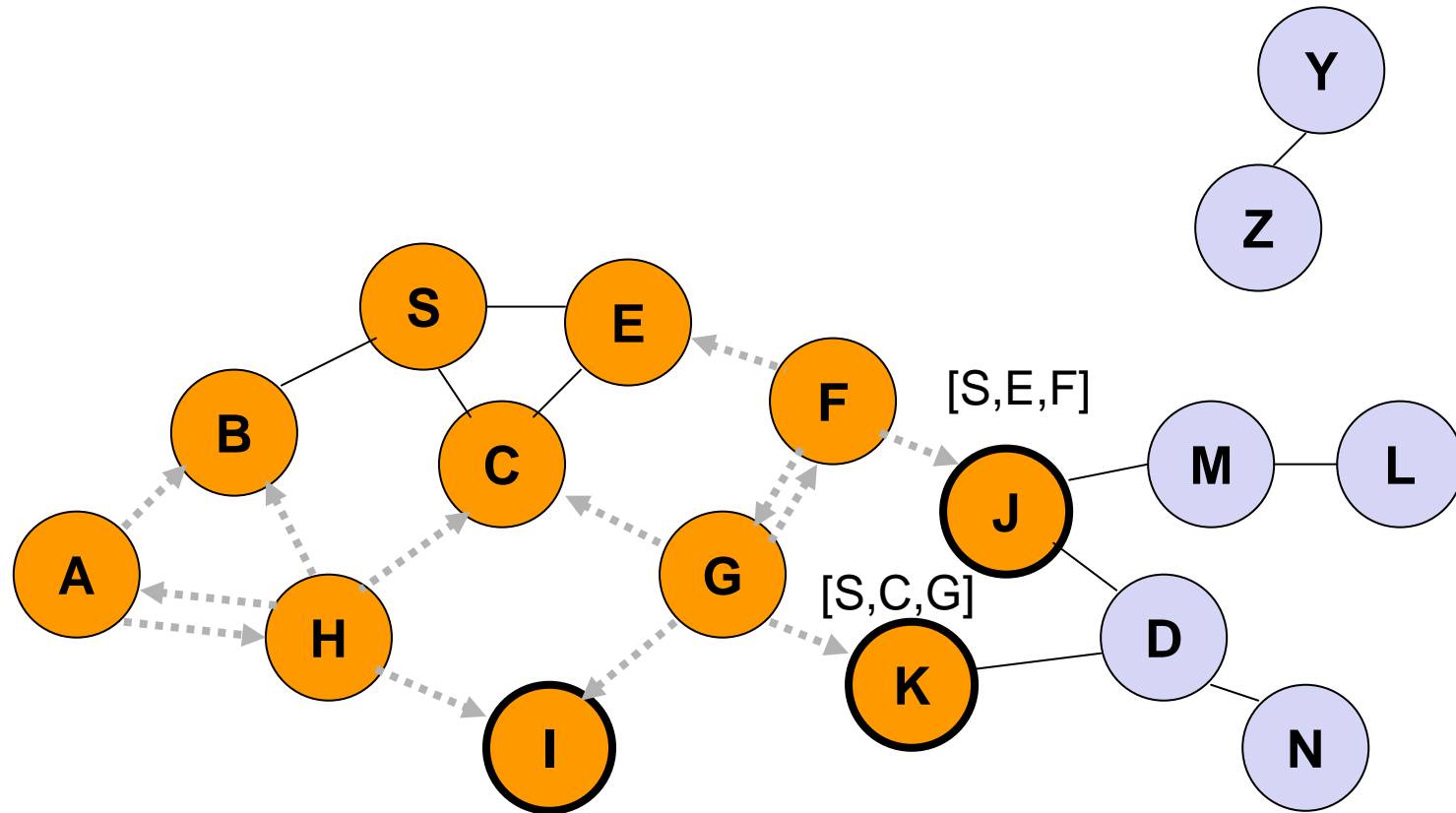
Beispiel DSR (3)



- ❑ Jeder Knoten, der Paket empfängt, leitet es per Broadcast weiter
- ❑ Eigene Adresse wird an RREQ angehängt

Quelle: N. Vaidya, Tutorial Mobicom 01

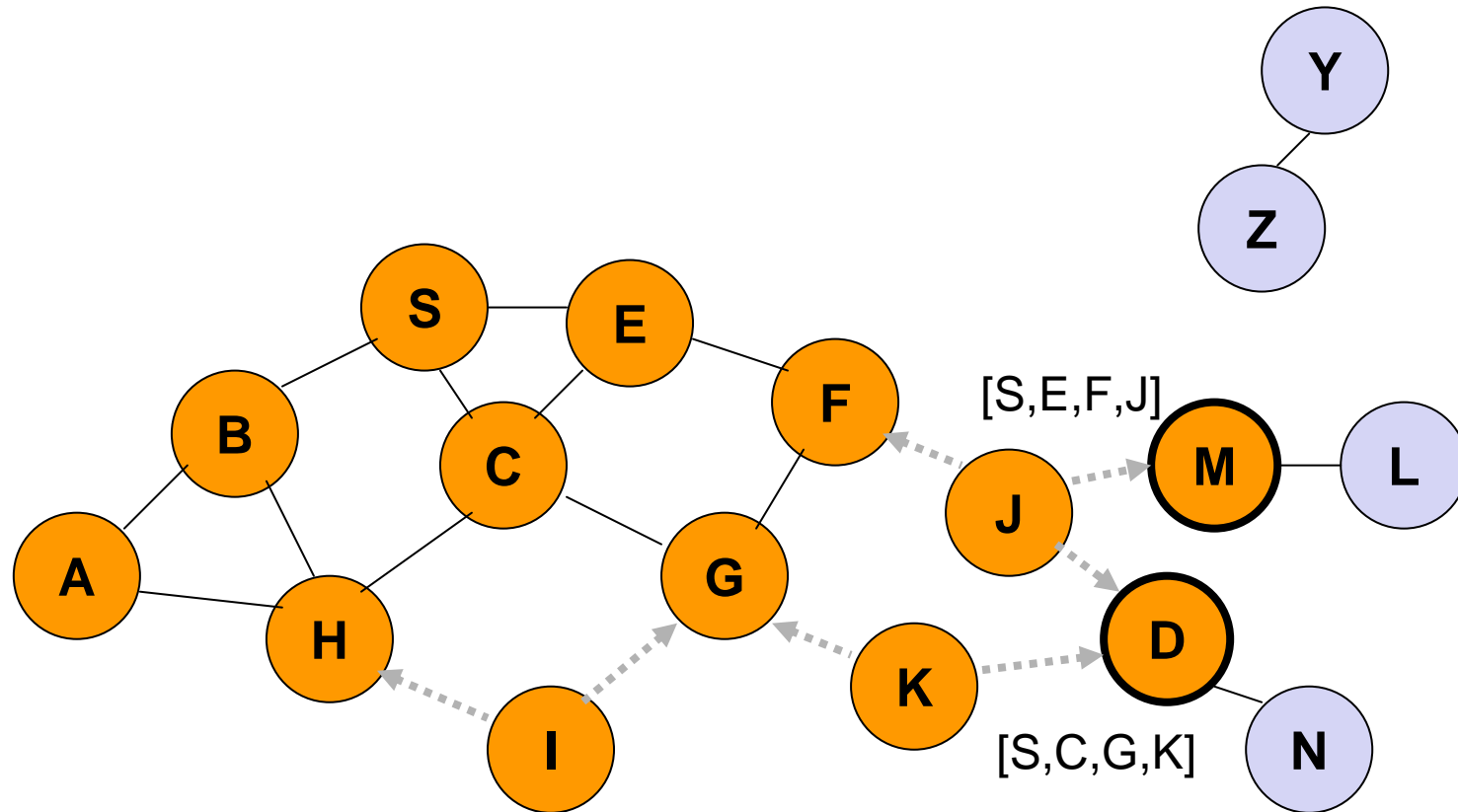
Beispiel DSR (4)



- ❑ C leitet Paket nicht ein zweites Mal weiter
- ❑ Sequenznummern verhindern, dass ein Knoten das gleiche Paket mehrfach weiterleitet

Quelle: N. Vaidya, Tutorial Mobicom 01

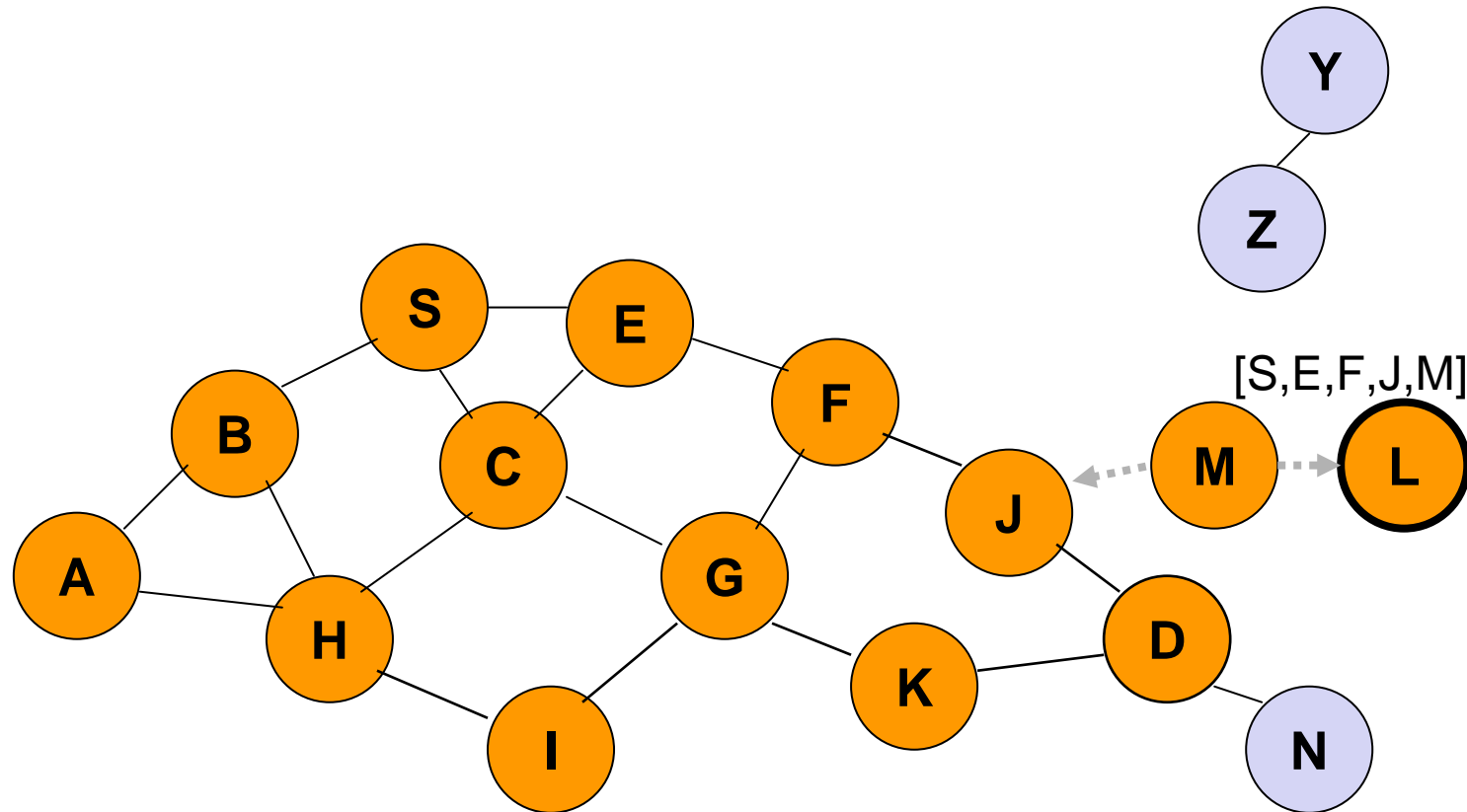
Beispiel DSR (5)



- ❑ Bei D könnte Paket von J und K kollidieren
- ❑ Falls MAC-Schicht keine Übertragungswiederholungen ausführt, wird RREQ nicht ankommen und müsste erneut von S versendet werden

Quelle: N. Vaidya, Tutorial Mobicom 01

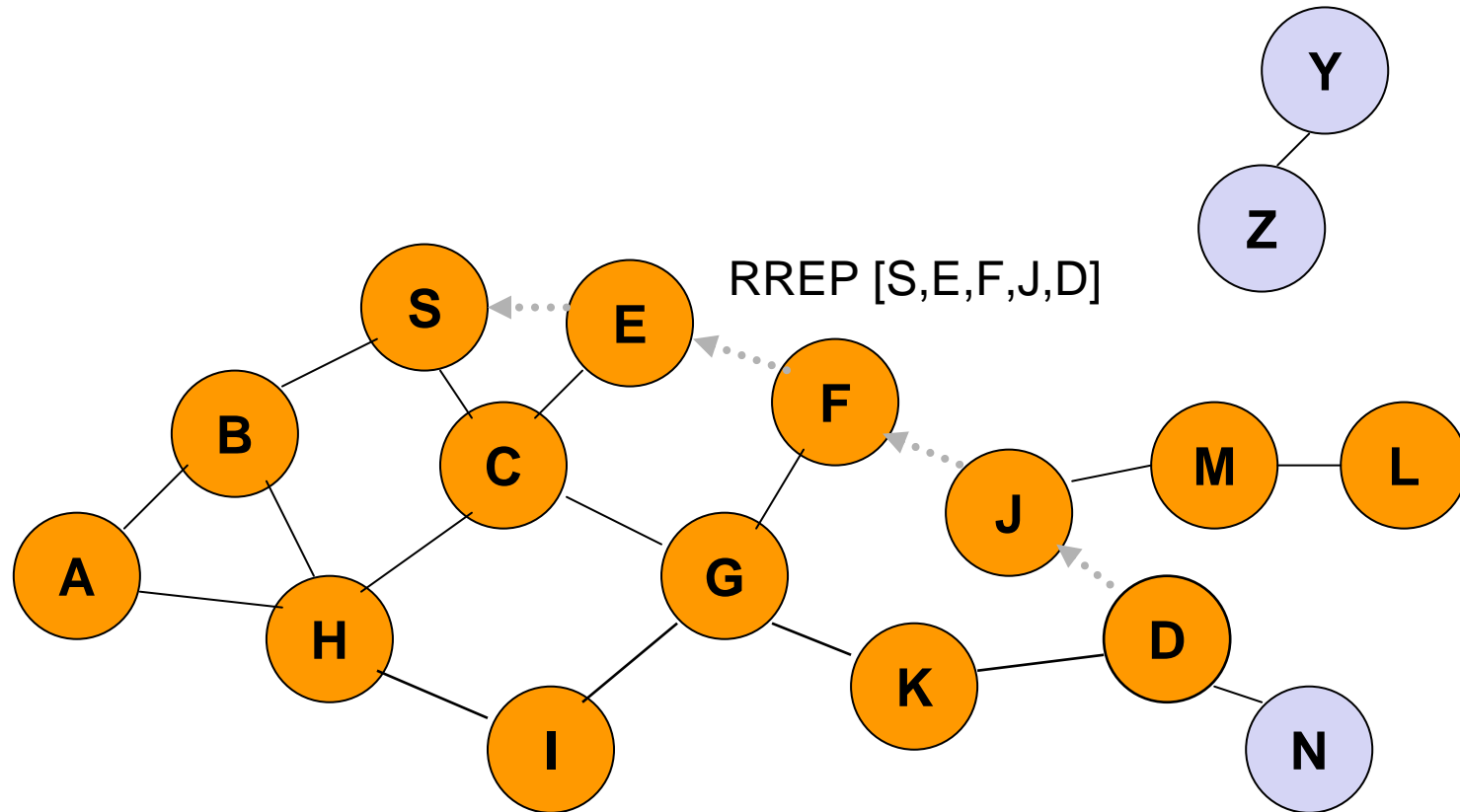
Beispiel DSR (6)



- Zielknoten D empfängt das Paket und leitet es nicht weiter

Quelle: N. Vaidya, Tutorial Mobicom 01

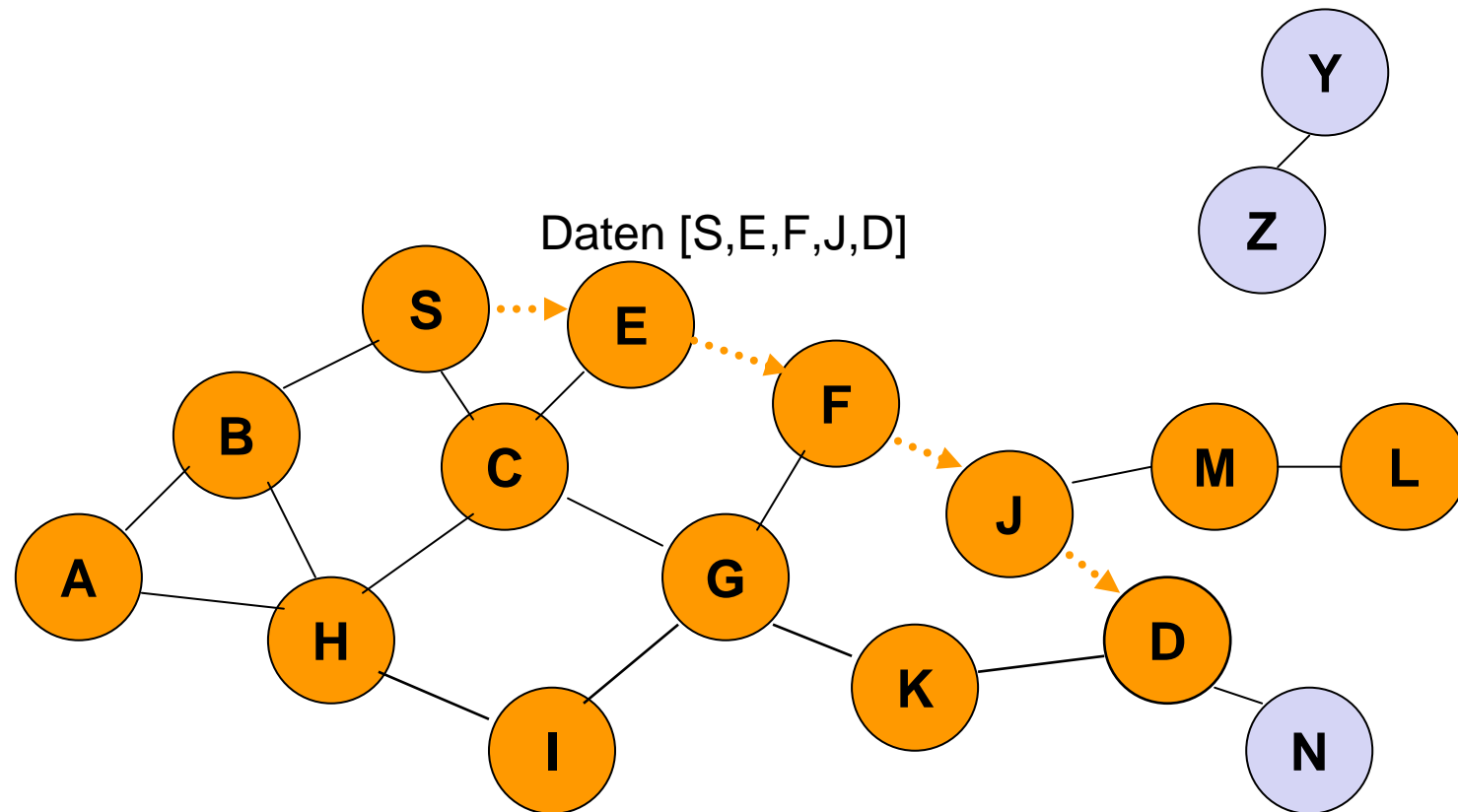
Beispiel DSR (7)



- ❑ Route Replay (RREP) enthält Weg von S zu D
- ❑ Falls MAC unidirektionale Links zulässt, muss D ein RREQ zu S starten und den RREP anhängen (hier nicht angenommen)

Quelle: N. Vaidya, Tutorial Mobicom 01

Beispiel DSR (8)



- ❑ Daten können über die Source Route [D,J,F,E,S] gesendet werden
- ❑ Größe des IP-Kopfs steigt mit Anzahl Hops

Quelle: N. Vaidya, Tutorial Mobicom 01

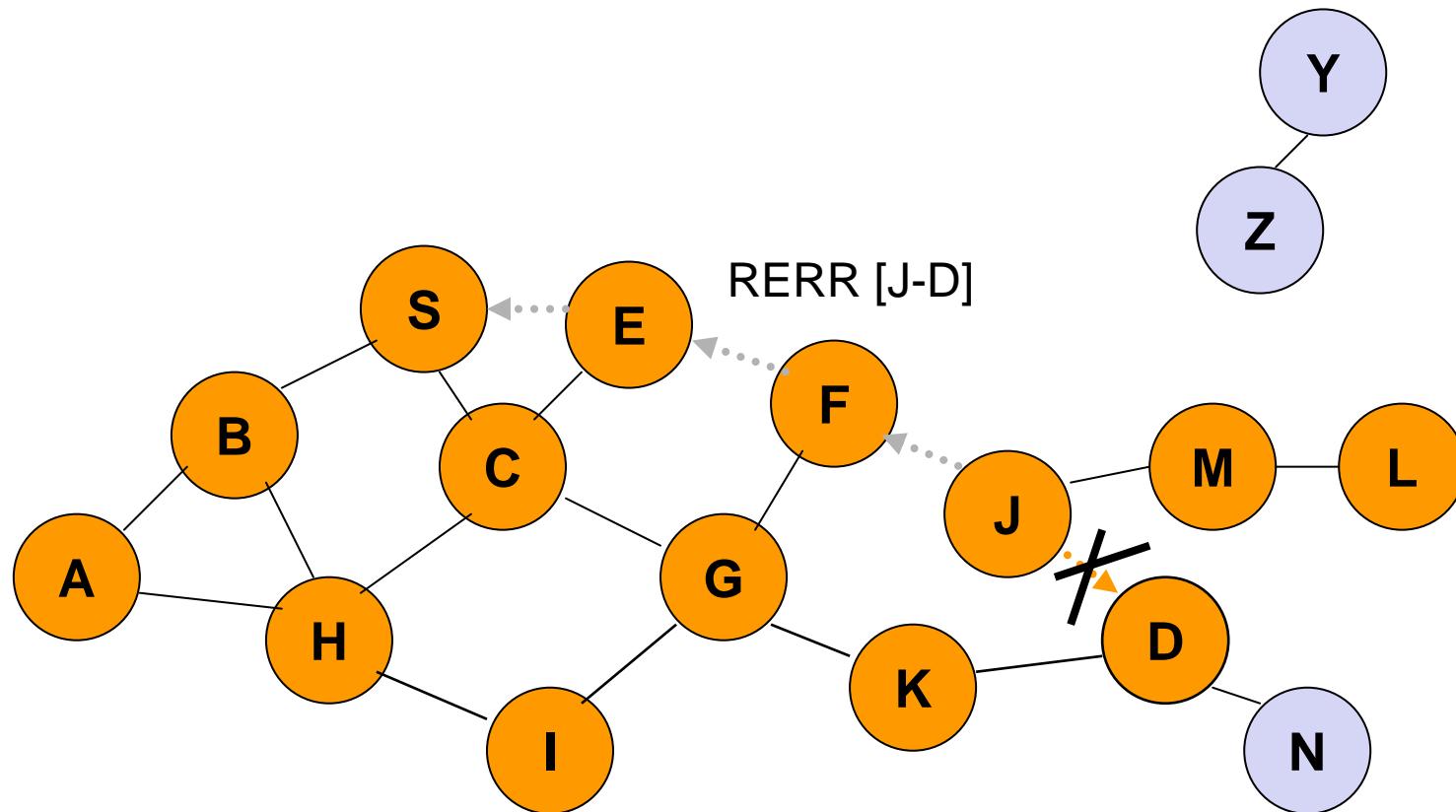
Nochmal: Auffinden eines Pfads bei DSR

- RREQ wird beim Sender ggf. nach Timeout wiederholt
- Dabei Verwendung eines exponentiellen Backoff, um den Overhead zu limitieren.
- Optimierungen
 - Begrenzung des Flutens auf Durchmesser des mobilen Netzes (=Hop Limit)
 - Ggf. *Expanding Ring Search* zum Ermitteln des Durchmessers
 - Caching von Weginformationen aus vorbeikommenden Paketen
 - Sender kann Cache nutzen um RREQ zu sparen
 - Knoten auf dem Weg zum Ziel können vorzeitig auf RREQ antworten mit Information aus eigenem Cache

Aufrechterhalten eines Wegs bei DSR

- Nach dem Senden
 - Warten auf die Quittung auf Schicht 2 (falls vorhanden, z.B. bei IEEE 802.11)
 - Mithören im Medium, ob Paket weitergeleitet wird (falls möglich)
 - Anforderung einer expliziten Bestätigung auf DSR-Ebene
 - Quittung kann über einen anderen Weg laufen
 - Wiederholen des Pakets, falls obige Maßnahmen nicht erfolgreich
 - Nach maximaler Anzahl Wiederholungen wird **Route Error** gesendet
 - Übertragungsabschnitt wird aus dem Cache genommen
- Falls Probleme erkannt werden kann der Sender informiert oder lokal ein neuer Weg gesucht werden

Beispiel DSR Route Error

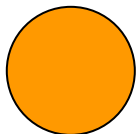
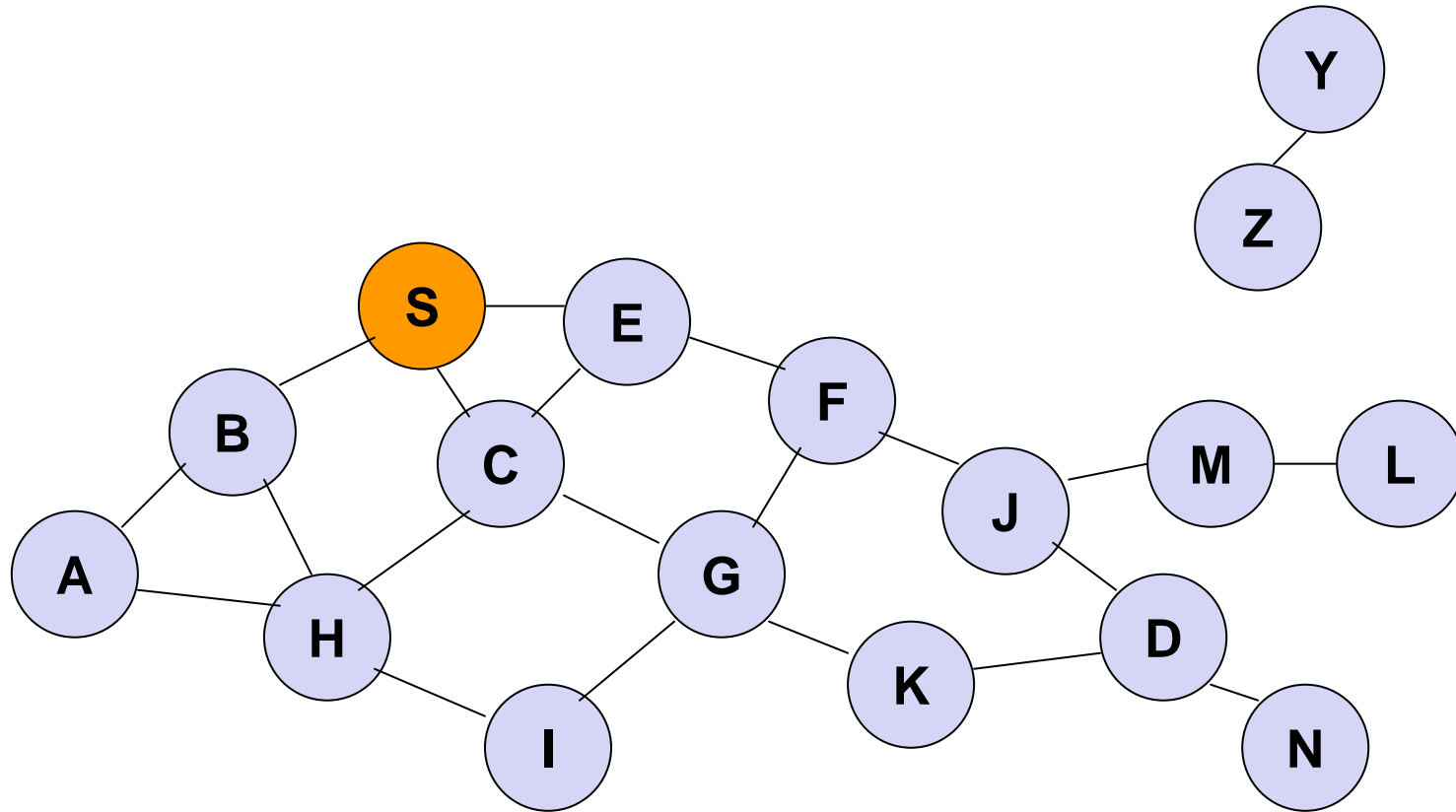


- ❑ J sendet Route Error (RERR) zu S, weil er Paket von S nicht weiterleiten konnte
- ❑ E und F entfernen den Link J-D aus ihrem Route Cache

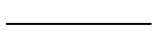
Quelle: N. Vaidya, Tutorial Mobicom 01

- Mischung zwischen DSR und DSDV
- Source Routes bei DSR führen zu großen RREQ/RREP-Paketen
 - AODV benutzt keine Source Routes
 - Stattdessen zusätzliche Informationen in Routingtabellen
- Bei Weiterleitung eines RREQ wird ein **Reverse Path** in die entgegengesetzte Richtung aufgebaut
 - Funktioniert nur bei bidirektionalen Links!
- RREP kann dann den vom RREQ aufgebauten Reverse Path verwenden
- Durch den RREP wird wiederum ein Reverse Path, der **Forward Path**, aufgebaut, der dann für den Datenverkehr verwendet werden kann

AODV Route Request (1)



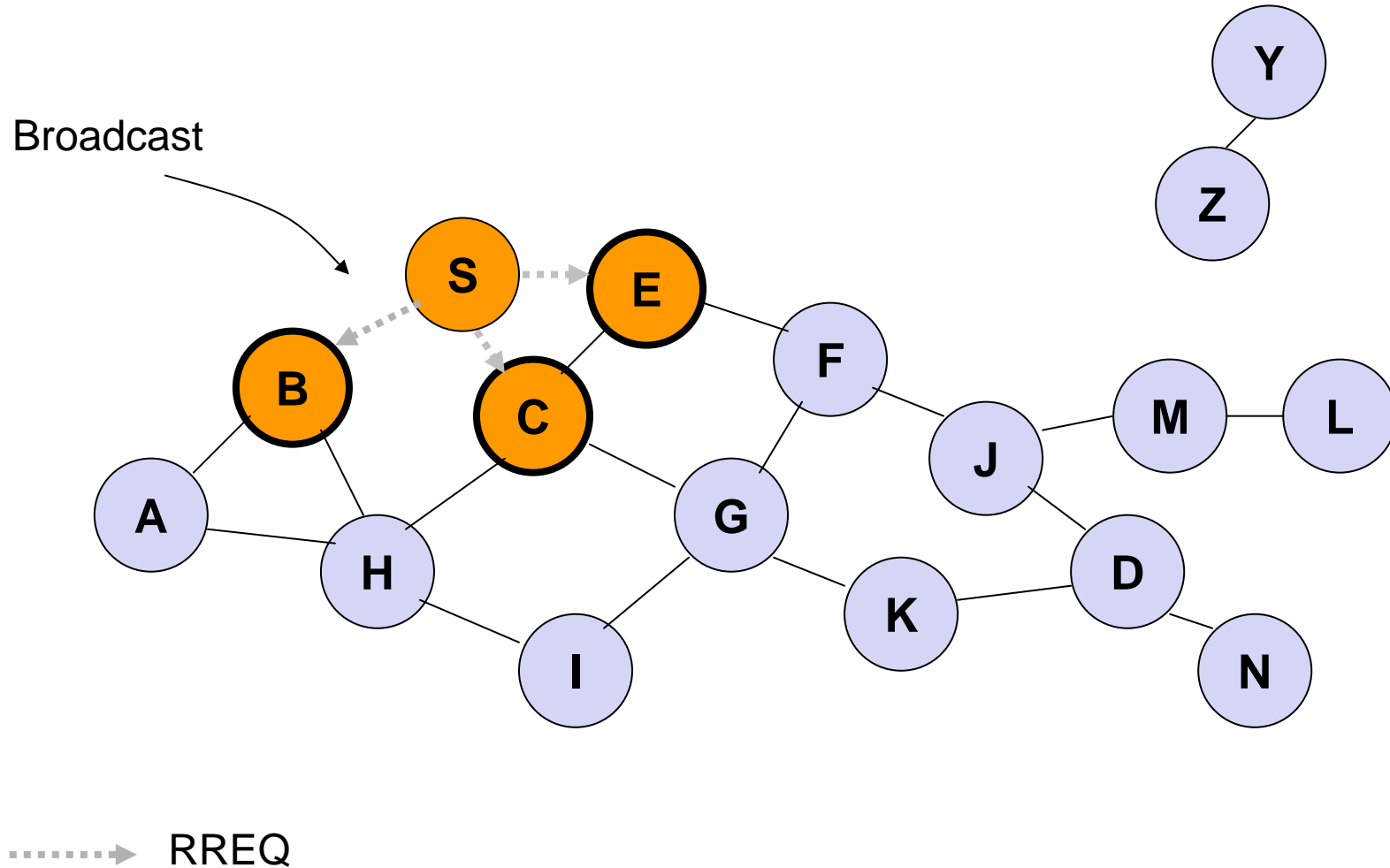
Knoten, der Route Request (RREQ) empfangen hat



Knoten in Sende/Empfangsreichweite

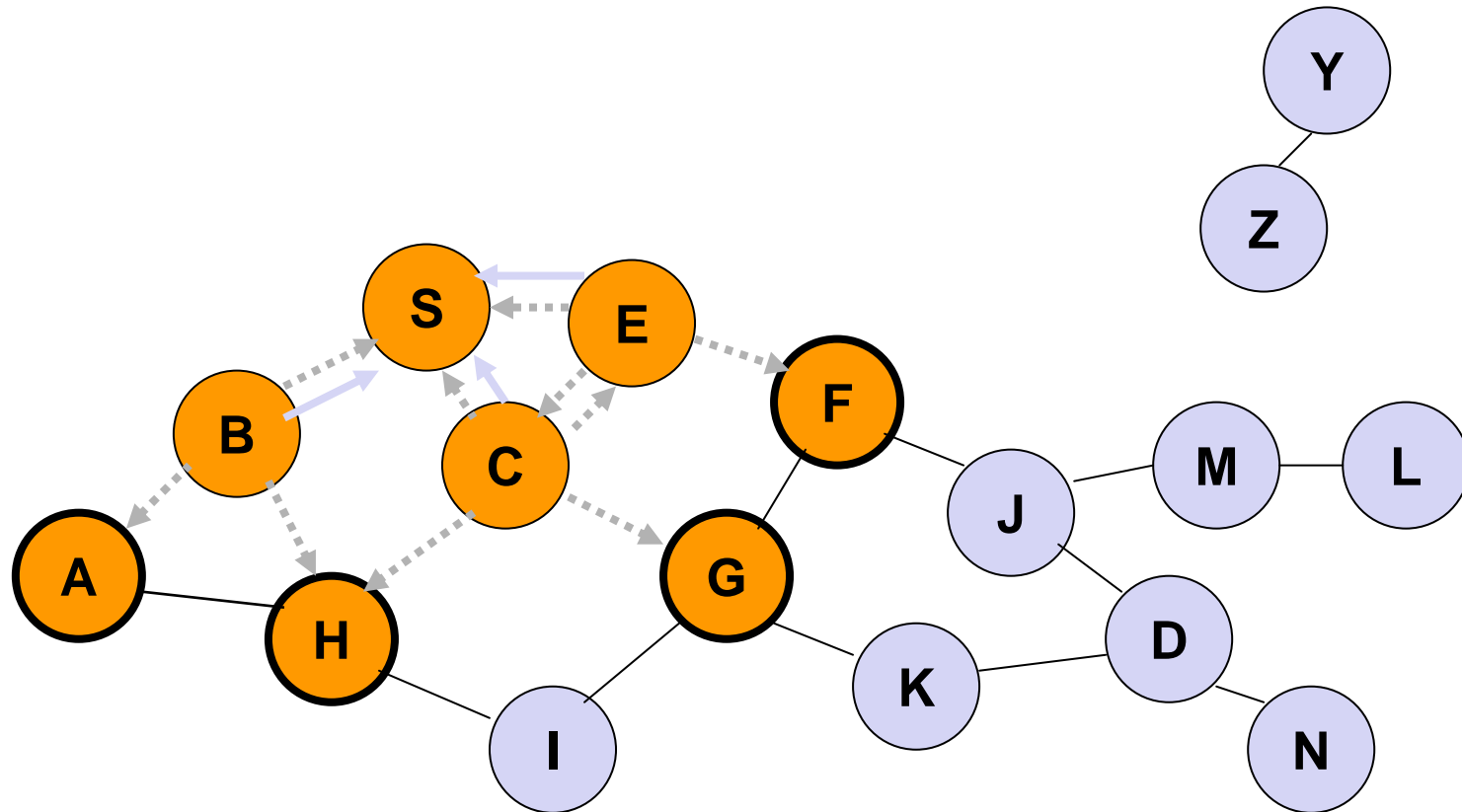
Quelle: N. Vaidya, Tutorial Mobicom 01

AODV Route Request (2)



Quelle: N. Vaidya, Tutorial Mobicom 01

AODV Route Request (3)

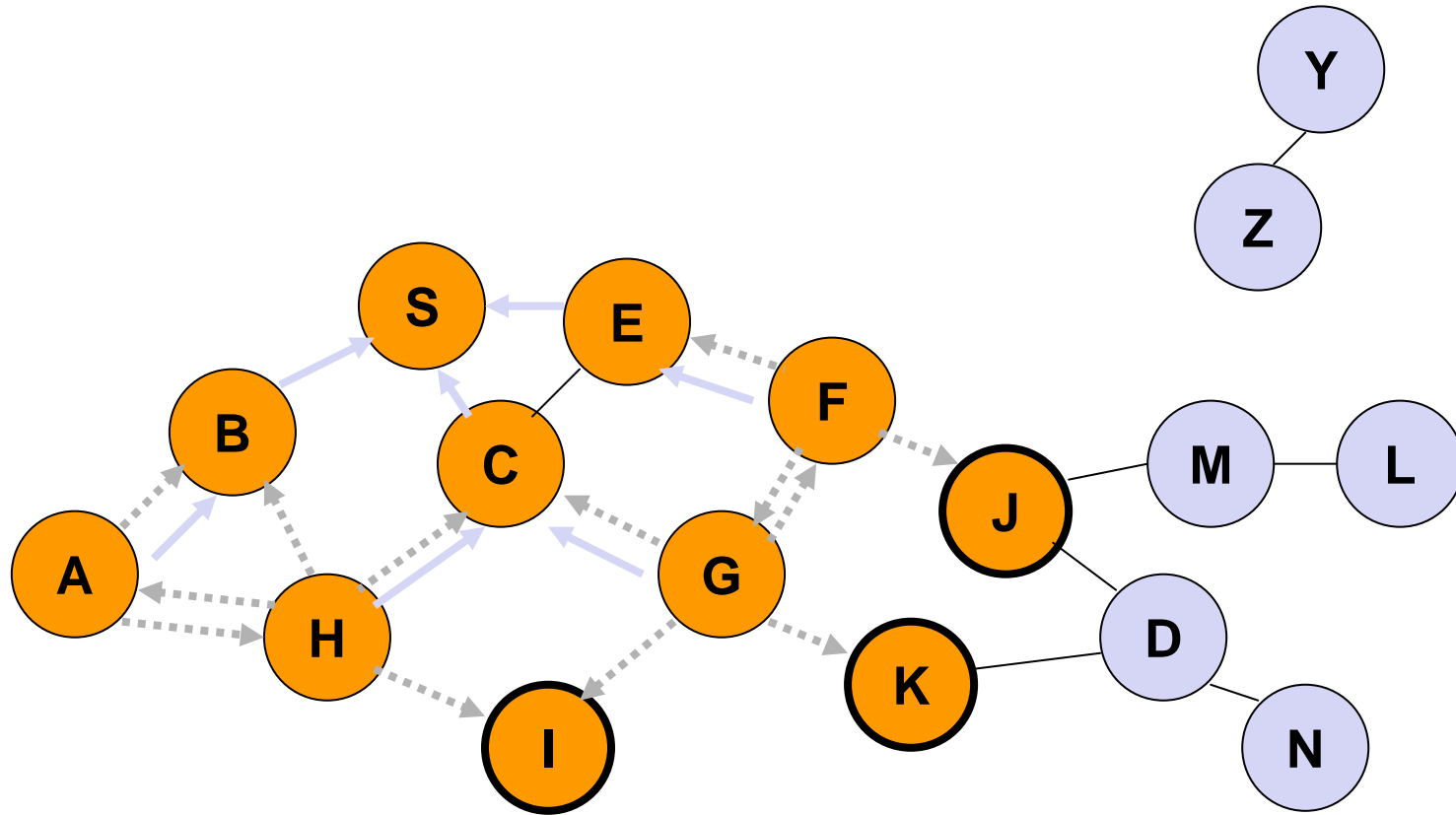


← Reverse Path

- Jeder Knoten merkt sich, von welchem Nachbar ein RREQ-Paket empfangen wurde → Zustandshaltung

Quelle: N. Vaidya, Tutorial Mobicom 01

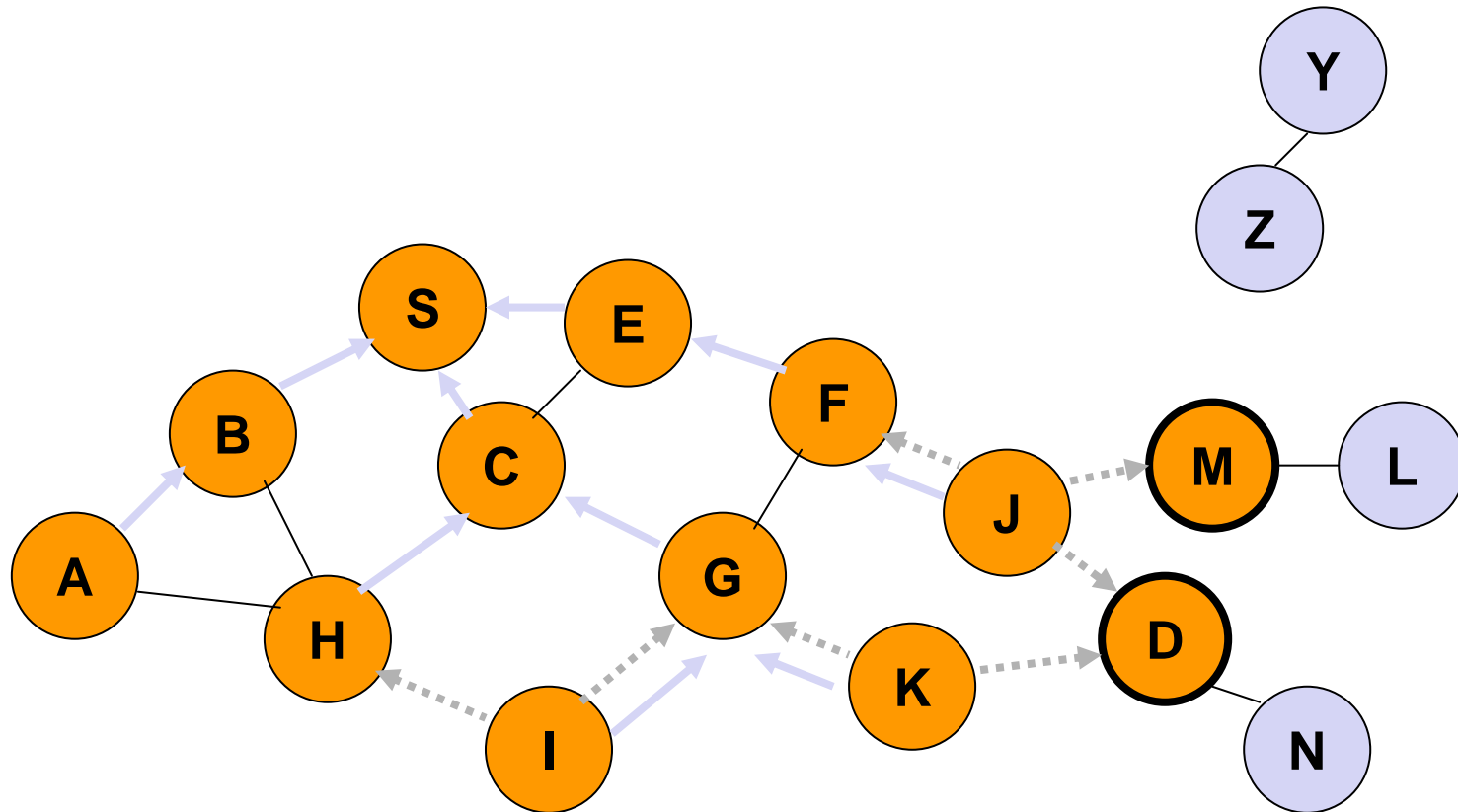
AODV Route Request (4)



- Dadurch baut sich ein Reverse Path von jedem Zwischenknoten zum Sender auf

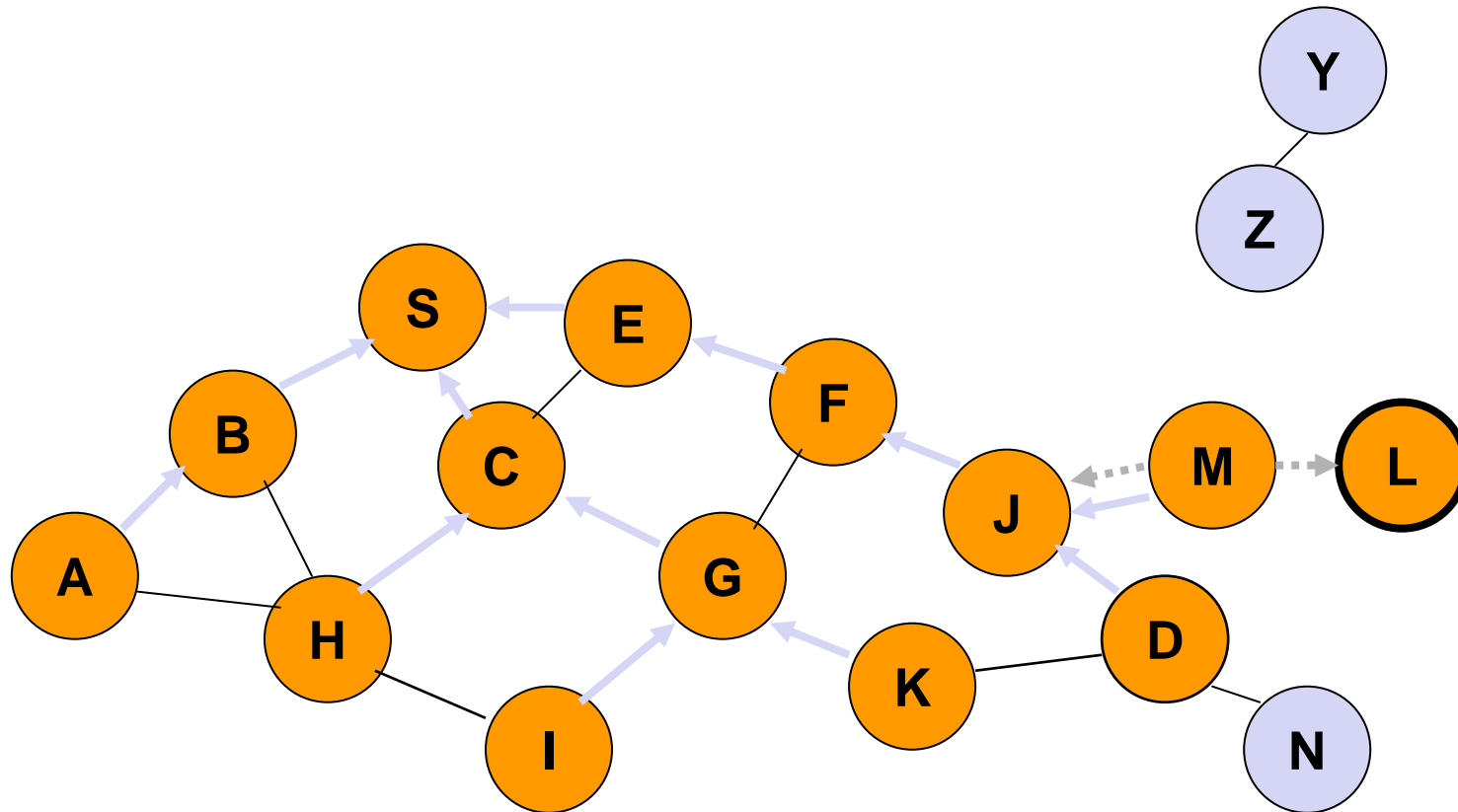
Quelle: N. Vaidya, Tutorial Mobicom 01

AODV Route Request (5)



Quelle: N. Vaidya, Tutorial Mobicom 01

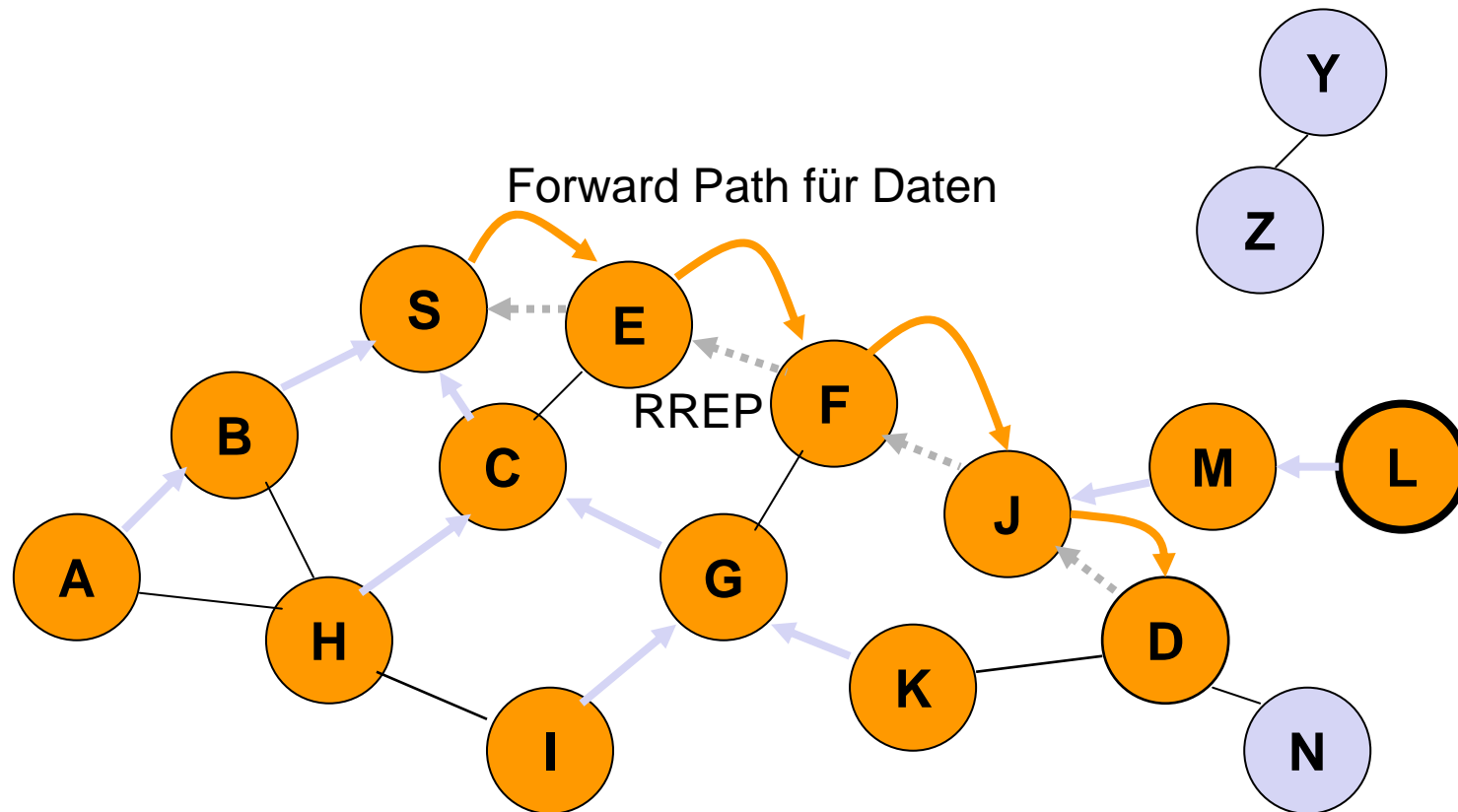
AODV Route Request (6)



- Wenn der RREQ bei D angekommen ist, existiert ein Reverse Path von D zu S

Quelle: N. Vaidya, Tutorial Mobicom 01

AODV Route Reply



- ❑ D sendet daraufhin RREP über Reverse Path zu S
- ❑ Durch den RREP wird wiederum ein Reverse Path aufgebaut, der den Forward Path für den Datenverkehr bildet
- ❑ Datenpakete enthalten *keine Source Route*

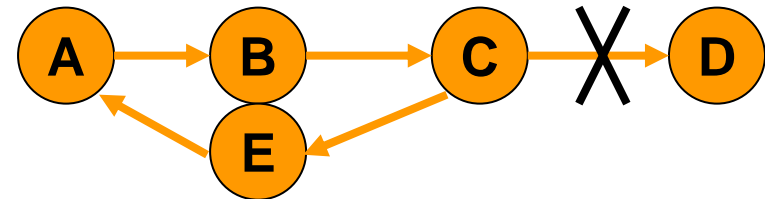
Quelle: N. Vaidya, Tutorial Mobicom 01

Sequenznummern und Route Error

- Reverse/Forwarding Path Einträge werden nach Timeout gelöscht → Softstate
- Jedem RREQ wird eine destination sequence number zugeordnet
 - Zum Erkennen von alten, nicht mehr gültigen Routen
 - Falls Zwischenknoten eine neuere Route kennen, können sie mit RREP antworten

– Verhindert Schleifen! - Beispiel:

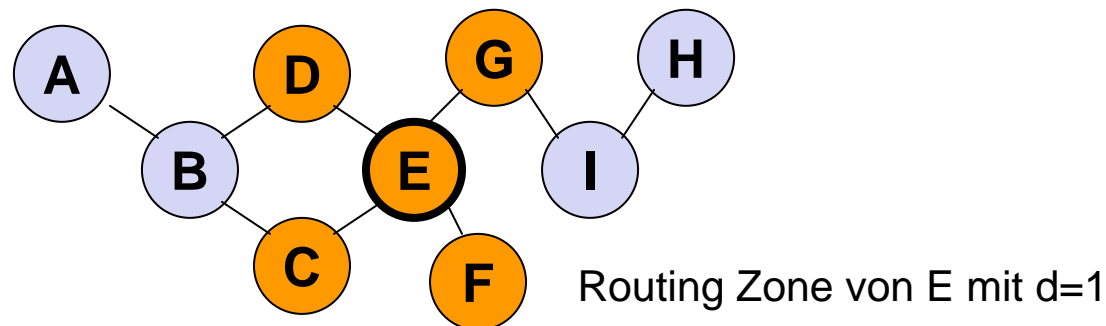
- RERR von C geht verloren
- C sendet RREQ
- A empfängt RREQ über C-E-A und antwortet mit RREP



- Falls ein Link nicht mehr zur Verfügung steht, werden alle aktiven Nachbarn mit RERR informiert
 - Nachbar ist aktiv, wenn er kürzlich ein Paket verschickt hat (active_route_timeout)
 - Link Fehler werden durch Hello-Messages oder Link-layer Acknowledgements erkannt
 - RERR erhält höhere destination sequence number
 - RERR wird so bis zur Quelle fortgepflanzt

Hybride Routingprotokolle

- Beispiel: Zone Routing Protocol (ZRP)
- Alle Knoten innerhalb einer Distanz von d Hops von Knoten X sind in der **Routing Zone** von Knoten X
- Routing Zonen der einzelnen Knoten überlappen sich
- **Intra-zone Routing** ist proaktiv, d.h. Routen zu allen Knoten innerhalb der Zone sind aus Routingtabelle bekannt.
- **Inter-zone Routing** ist reaktiv. Dazu wird ein RREQ zu allen Knoten *am Rand der Zone* geschickt, die dann entweder den Zielknoten kennen (proaktiv) oder wiederum ein Route Discovery ausführen müssen (reaktiv).



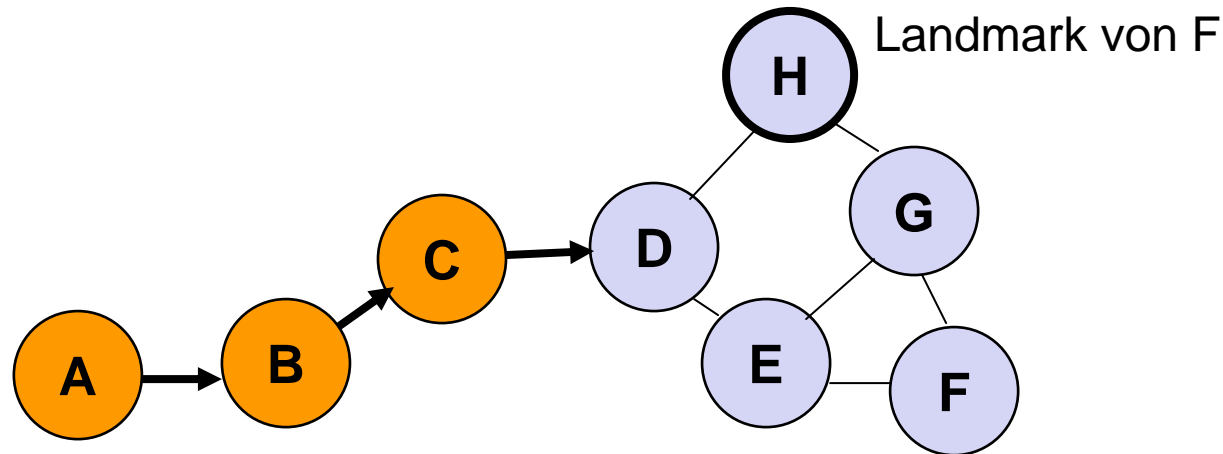
Quelle: N. Vaidya, Tutorial Mobicom 01

Hierarchische Routingprotokolle

- Bereitstellung einer geeigneten selbstorganisierenden Kontrollstruktur für große dynamische Netze
- Beispiel: Landmark Routing (LANMAR)
 - Annahme: Gruppe von Knoten, die sich größtenteils im Verbund bewegen, bilden logisches Subnetz
 - In jedem Subnetz wird ein **Landmark** Knoten gewählt
 - Jeder Knoten besitzt eigenen **Scope**, definiert als Menge aller Knoten, die nicht weiter als r Hops entfernt sind
 - Jeder Knoten leitet Link Status Informationen von Knoten innerhalb seines Scopes und Distanz Vektor Informationen aller Landmarks weiter
 - Knoten kennen nur Routen zu Knoten innerhalb des eigenen Scopes
 - Alle Knoten kennen nächsten Hop zu allen Landmarks
 - Pakete außerhalb des Scopes werden in Richtung des zum Zielknoten gehörenden Landmarks geschickt
 - Ermittelt durch Subnetz-Teil der hierarchischen Adresse
 - Wenn Paket in Scope des Zielknoten eintritt, wird es direkt zum Zielknoten weitergeleitet
 - Route muss nicht durch Landmark Knoten gehen!

Beispiel LANMAR (1)

- Routing innerhalb des Scopes
 - Knoten A möchte Knoten F erreichen
 - Knoten A sendet Datenpakete Richtung H
 - Knoten A, B und C kennen nächsten Hop zu Knoten H auf Grund erhaltener Distanz Vektor Informationen

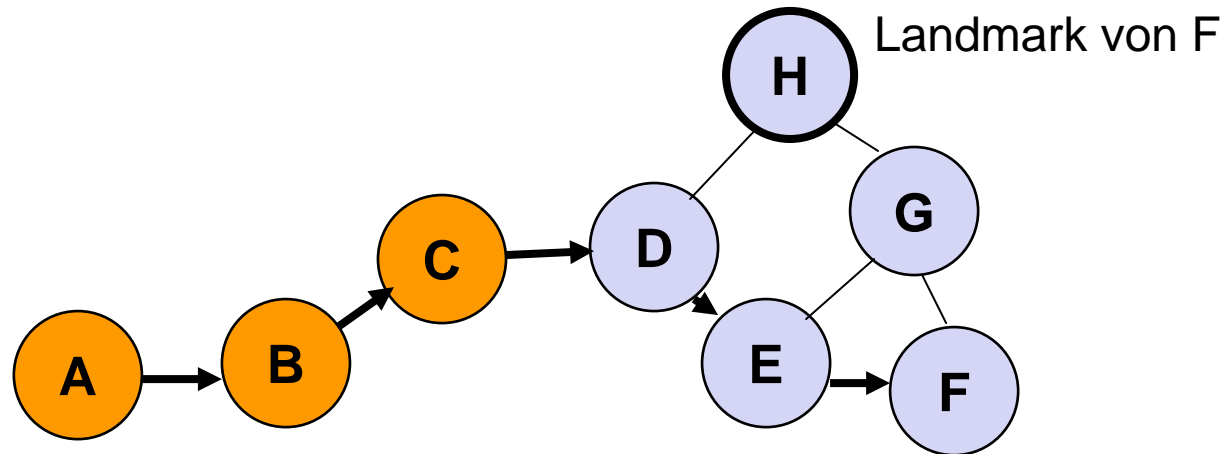


- Knoten gehört zum Scope von Knoten A
- Knoten gehört zum Scope von Knoten F

Quelle: N. Vaidya, Tutorial Mobicom 01

Beispiel LANMAR (2)

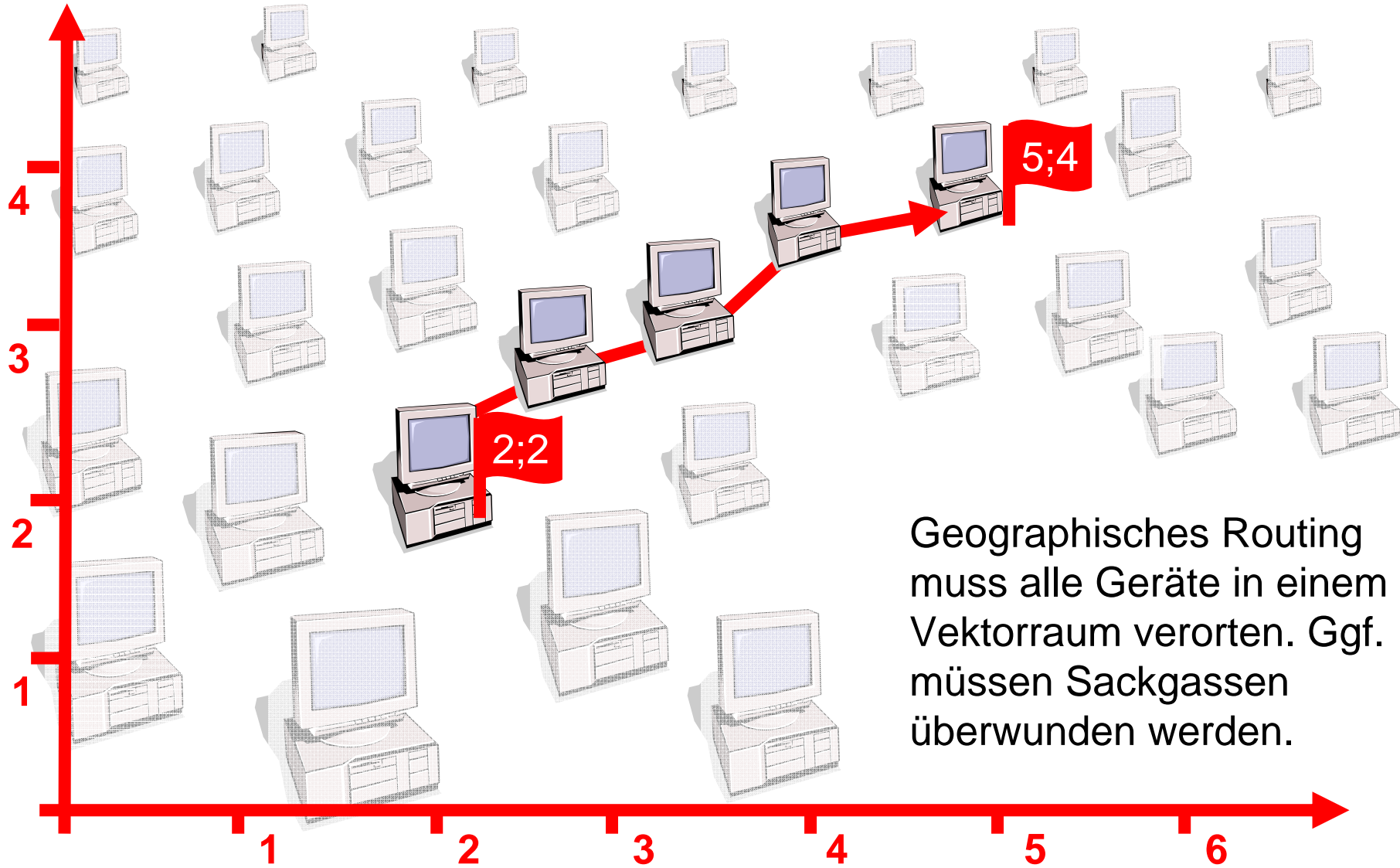
- Routing außerhalb des Scopes
 - Knoten D ist im Scope von Knoten F
 - Daher kennt Knoten D nächsten Hop zu F und kann Paket direkt zu F weiterleiten
 - Route muss nicht Landmark (Knoten H) enthalten



- Knoten gehört zum Scope von Knoten A
- Knoten gehört zum Scope von Knoten F

Quelle: N. Vaidya, Tutorial Mobicom 01

Ausblick: Geographisches Routen



Geographisches Routing muss alle Geräte in einem Vektorraum verorten. Ggf. müssen Sackgassen überwunden werden.

- Routing protocols from the fixed wired Internet must be adapted to work in wireless and mobile settings
- DSDV: Introduce sequence numbers to avoid routing loops (=, 'count to infinity' problem)
- OLSR: Not all routers relay topology information
- DSR: Flood the network with discovery packets; reply carries source route to the destination
- AODV: Flood the network and build up state while flooding and replying
- Zone routing: Route differently in the vicinity of the sender
- Landmark routing: Have different granularity of routing information depending on distance from sender
- Geographic routing: Use physical coordinates as destination address

Questions?



Thomas Fuhrmann

Department of Informatics
Self-Organizing Systems Group
c/o I8 Network Architectures and Services
Technical University Munich, Germany

fuhrmann@net.in.tum.de