

Kapitel 10 – Mobile Ad Hoc Netze

Vorlesung Mobilkommunikation Wintersemester 2016/17
Prof. Dr. Oliver Waldhorst (HS Karlsruhe), Markus Jung

INSTITUT FÜR TELEMATIK





Mobiles TCP



Mobile Ad Hoc Netze



Mobile IP



WLAN, Bluetooth



GSM, UMTS, LTE



Mobilitätsmanagement



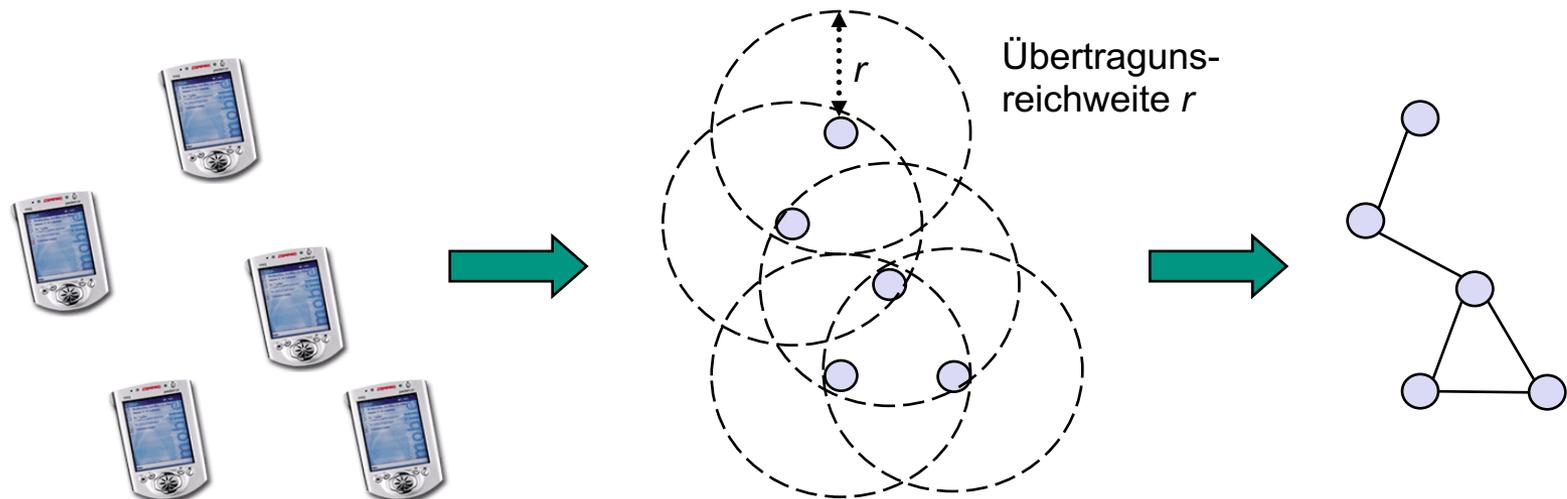
Medienzugriff



Drahtlose Übertragung

Mobile Ad Hoc Netze (MANETs)

- Spontaner Zusammenschluss drahtloser Endgeräte
 - Keine Infrastruktur (Basisstation/Access Points), kein Backbone
 - Verwendete Endgeräte können mobil sein
- Paketbasierte Vermittlung von Daten
 - Routen zwischen zwei Geräten können *mehrere Hops* lang sein
- Jedes Gerät ist Endgerät und gleichzeitig *Router!*
- Ad Hoc Netze sind *selbstorganisierend*
 - Keine zentralen Komponenten



Vorteile und Anwendungen

- Vorteile mobiler Ad Hoc Netze
 - Netze können einfach, kostengünstig und schnell aufgebaut werden
 - Z.B. mit 802.11b im lizenzfreien Band (2.4 GHz)
 - Sendeleistung kann reduziert werden
 - Entfernung bis zu Nachbarn evtl. geringer als zur Basisstation
 - Robuster gegenüber dem Ausfall einzelner Komponenten da vollständig dezentral

- Anwendung: Überall wo kein Zugriff auf Infrastruktur besteht
 - Militärischer Bereich
 - Verbände von Soldaten, Panzern, Flugzeugen,...
 - Ziviler Bereich
 - Konferenzen, Ausstellungen, Meetings, Vorlesungen, Gaming
 - Car-to-Car-Kommunikation, Netz für Taxifahrer, Polizeistreifen
 - Erweiterung zellularer Systeme (WLAN, UMTS)
 - Katastropheneinsätze
 - Nach Zusammenbruch der Infrastruktur (Telefonnetz bei Erdbeben)
 - Rettungsaktionen z.B. nach Lawinenunglücken

Eigenschaften von MANETs

- Hoch **dynamische** Netztopologie
 - Mobilität der Geräte
 - Sich verändernde Eigenschaften des drahtlosen Kanals (Fading)
 - Partitionierung und Zusammenschlüsse von Ad Hoc Netzen möglich
- **Asymmetrische/Unidirektionale** Verbindungen
 - Verbindungsqualität kann in beide Richtungen unterschiedlich sein
- Drahtloses Medium ist **Semi-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 Nachrichten an andere weiterleiten können



[10.1]

Routingprotokolle für MANETs

- Für Festnetz entwickelte Routingprotokolle (z.B. RIP, OSPF, ...) können nicht ohne weiteres in mobilen Ad Hoc Netzen angewendet werden
 - Langsame Konvergenz
 - Zu hoher Overhead
- Routingprotokolle für MANETs müssen dagegen schnell konvergieren und möglichst wenig Bandbreite für den Kontrollverkehr verwenden
- Verschiedene Metriken in mobilen Ad Hoc Netzen möglich
 - Minimale Anzahl Hops
 - Minimaler Delay
 - Minimale Paketverluste
 - Minimale Stausituationen (Load Balancing)
 - Minimale Interferenzen
 - Maximale Signalstabilität und zeitlich stabile Route
 - Maximale Batterielaufzeit eines mobilen Gerätes
 - Maximale Lebenszeit des gesamten Netzes
 - z.B. bis Partitionierung durch Batterielaufzeit-bedingten Ausfall von Stationen

Übersicht Ad-hoc-Routingprotokolle

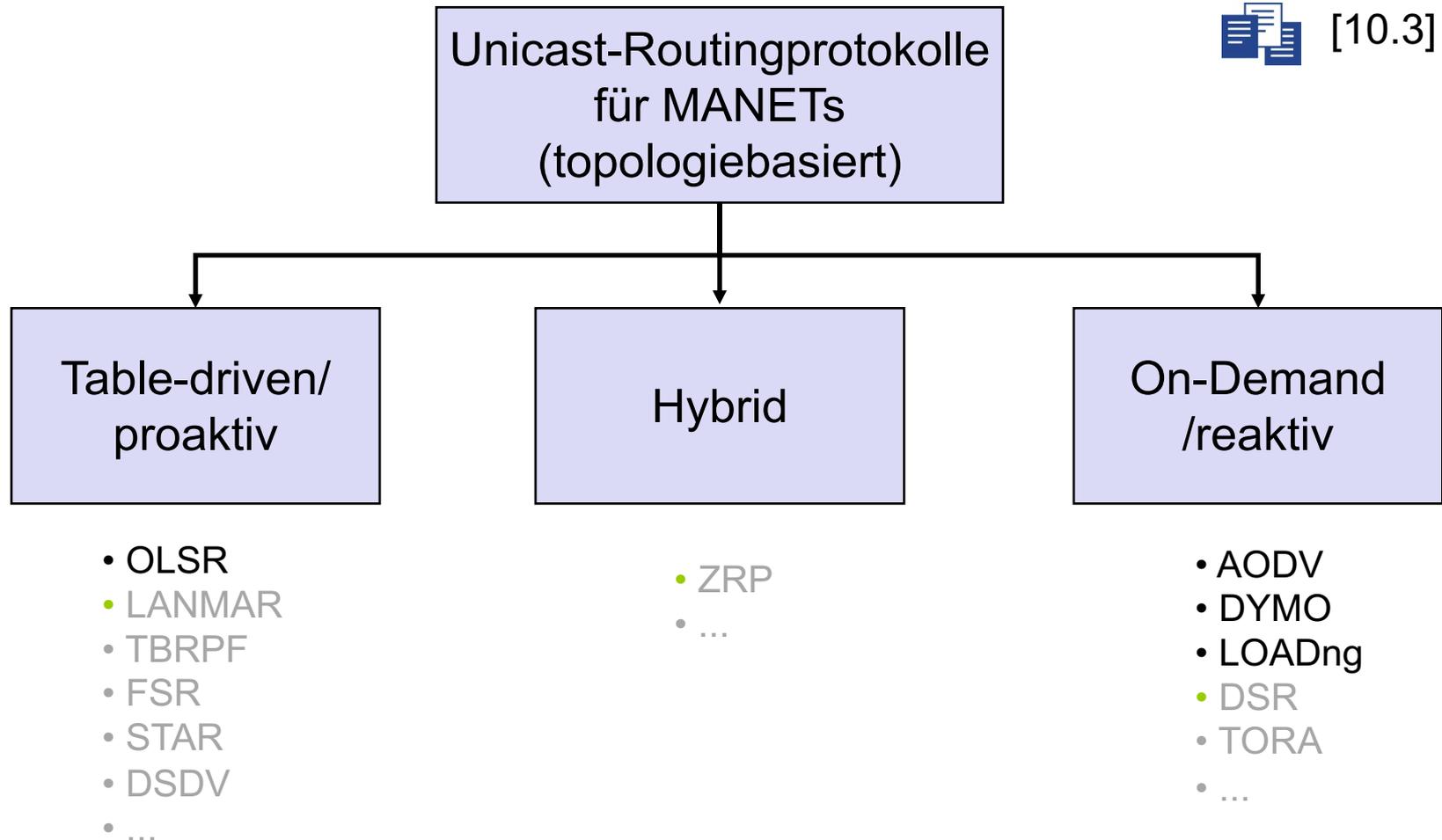
- **Fluten** für den Datenversand
 - Einfachstes „Protokoll“: Jede Station leitet jede Nachricht genau einmal weiter
 - Sehr hoher Overhead
- **Proaktives Routing** (Table-driven)
 - Es werden fortlaufend Routen zu allen anderen Stationen unterhalten
 - Routen stehen ständig zur Verfügung
 - Konstant hoher Kontrolloverhead
- **Reaktives Routing** (On-demand)
 - Routen werden nur bei Bedarf bestimmt (Route Discovery)
 - Zeitverzögerung zu Beginn, da zunächst Route bestimmt werden muss
 - Kontrolloverhead abhängig von Anzahl der Kommunikationsbeziehungen
- **Hybrides Routing**
 - Mischung aus proaktivem und reaktivem Routing

- Es gibt (bisher) nicht *das* Routingprotokoll für mobile Ad Hoc Netze
 - Je nach Szenario ist das eine oder andere besser geeignet
 - Zahlreiche Publikationen auf diesem Gebiet
- Standardisierung von Ad-hoc-Routingprotokollen: IETF MANET WG



[10.2]

Klassifikation von Routingprotokollen



Hier nicht behandelt: positionsbasierte Routingprotokolle

Abkürzungen für Routingprotokolle

- OLSR Optimized Link State Routing
- LANMAR Landmark Ad Hoc Routing
- TBRPF Topology Dissemination based on Reverse-Path Forwarding

- FSR Fisheye State Routing
- STAR Source Tree Adaptive Routing
- DSDV Destination-Sequenced Distance Vector
- ZRP Zone Routing Protocol
- DSR Dynamic Source Routing
- AODV Ad Hoc On Demand Distance Vector
- DYMO Dynamic MANET On-demand
- TORA Temporally-Ordered Routing Algorithm

Proaktive Routingprotokolle

- Station bestimmt fortlaufend Routen zu allen anderen Stationen im Netz
 - Distance Vector Routing
 - Basiert auf Bellman-Ford Algorithmus
 - Danach ermittelt jede Station jeweils für jeden anderen Station im Netz den Nachbarn (nächsten Hop), über den die kürzeste Route zu dieser Station besteht und die Länge dieser Route
 - Diese Information wird periodisch an alle Nachbarn versendet
 - Beispiele
 - Festnetz: Routing Information Protocol (RIP)
 - Ad Hoc Netz: Destination-Sequenced Distance Vector Protocol (DSDV)
 - Link State Routing
 - Jede Station versendet periodisch den Status seiner Links, zusammen mit den von Nachbarn empfangenen Link Status Informationen
 - Dadurch kennt jede Station gesamte Netztopologie
 - Kürzeste Route zu jeder Station kann mit Dijkstra's Shortest Path First Algorithmus ermittelt werden
 - Beispiele
 - Festnetz: Open Shortest Path First (OSPF)
 - Ad Hoc Netz: Optimized Link State Routing (OLSR)

Reaktive Routingprotokolle

- Prinzip
 - Station kennt nur die Routen, die er auch benötigt
 - Keine periodischen Aktualisierungen
- Aufgaben eines reaktiven Routingprotokolls
 - Auffinden einer Route (**Route Discovery**)
 - Nur wenn eine Station Daten zu einer Zielstation senden möchte, jedoch noch keine Route zu dieser Zielstation besitzt
 - Aufrechterhaltung einer Route (**Route Maintenance**)
 - Nur wenn eine Route verwendet wird, dann wird auch dafür gesorgt werden, dass diese auch weiterhin funktioniert
- Vergleich mit proaktiven Routingprotokollen
 - Vorteil
 - Nur verwendete Routen werden bestimmt und aufrecht erhalten
 - Kein periodisches Versenden von Nachrichten nötig → Ressourceneinsparung
 - Nachteil
 - Zeitverzögerung zu Beginn einer Kommunikation, da Route erst noch bestimmt werden muss
 - Kontrolloverhead abhängig von Anzahl der Kommunikationsbeziehungen und Mobilität

Ad-hoc On-Demand Distance Vector Routing Protokoll (AODV)

→ Auffinden einer Route

- Sender S flutet das Ad-hoc Netz mit einem Route Request (RREQ)
 - RREQ enthält Zieladresse Z
- Stationen, die an der Weiterleitung des RREQs beteiligt sind, speichern die Adresse der Station, von welchem das RREQ empfangen wurde
- Es entsteht der so genannte „Reverse Path“ in Richtung des Senders S
 - Funktioniert nur bei bidirektionalen Links!
- Erhält Ziestation Z den RREQ, beantwortet er dieses durch ein Route Reply (RREP)
 - RREP enthält Quelladresse S
- Weiterleitung des RREPs in Richtung von Station S ist direkt über den zuvor aufgebauten Reverse Path möglich
- Bei der Weiterleitung des RREPs wird der so genannte „Forward Path“ in Richtung der Zielstation Z aufgebaut
- Der aufgebaute Forward Path wird für den Datenverkehr verwendet

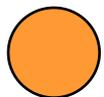
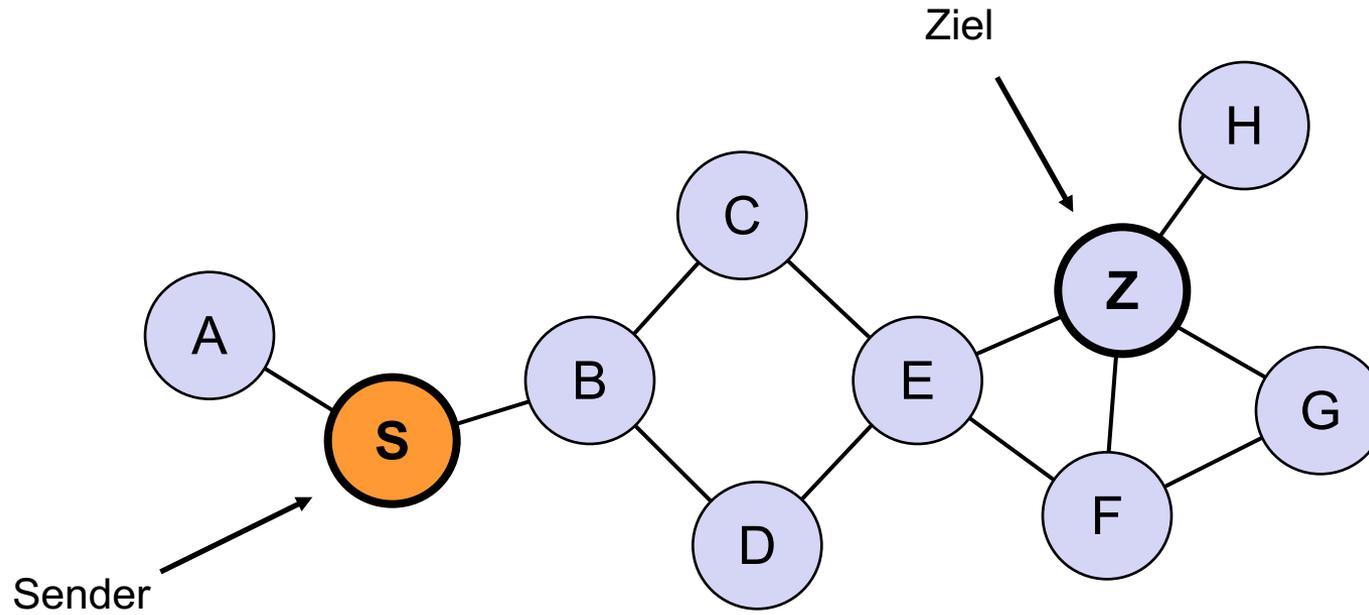
→ Wird eine Route von S nach Z aufgebaut, so existiert automatisch auch eine Route von Z nach S!



[10.4]

AODV: Fluten des RREQs I

Suche eines Pfades von S nach Z:

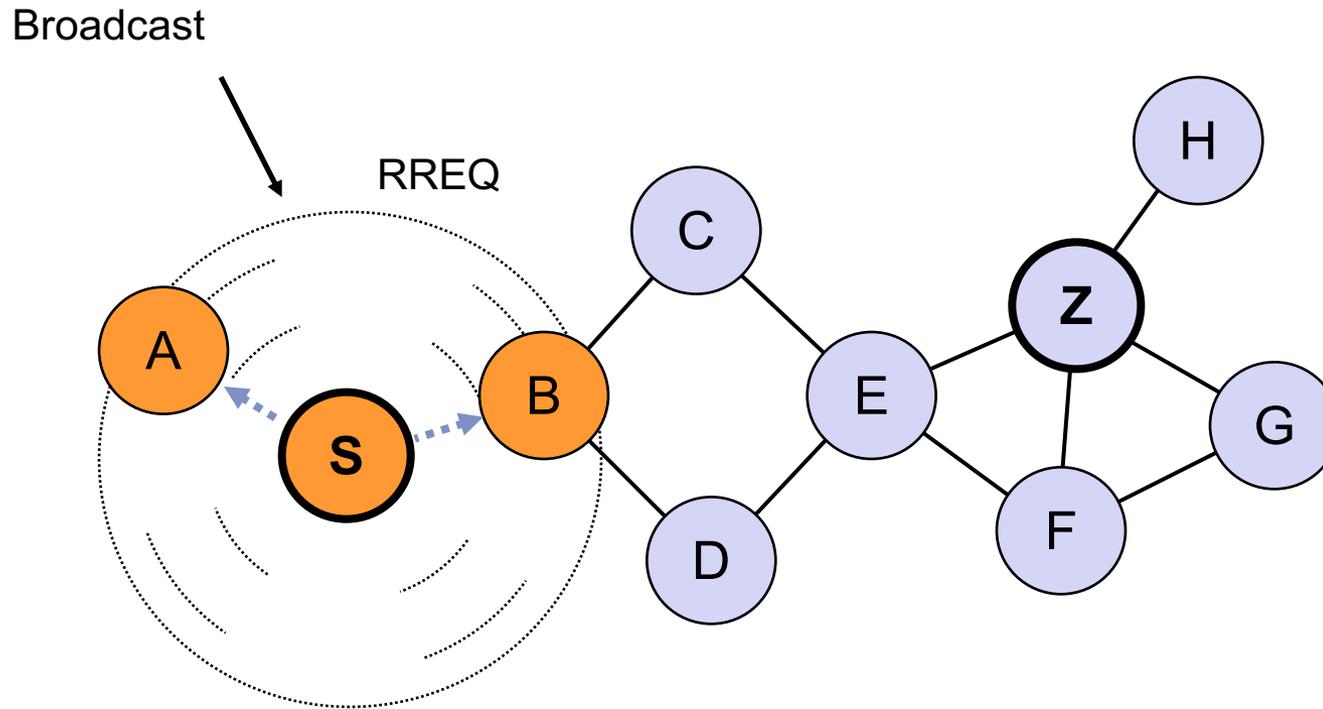


Station, die RREQ empfangen hat



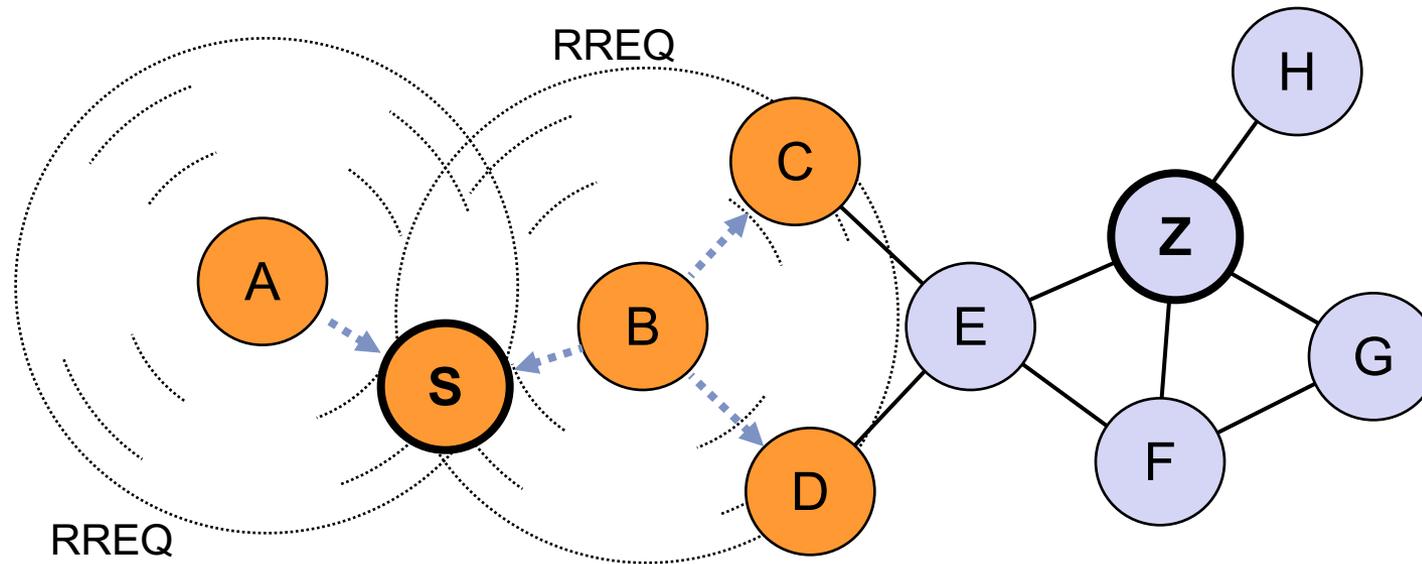
Stationen in Sende- und Empfangsreichweite (bidirektionale Links)

AODV: Fluten des RREQs II



- Sender S verschickt **RREQ** per **Broadcast**
 - Alle Nachbarn in Reichweite empfangen die Nachricht

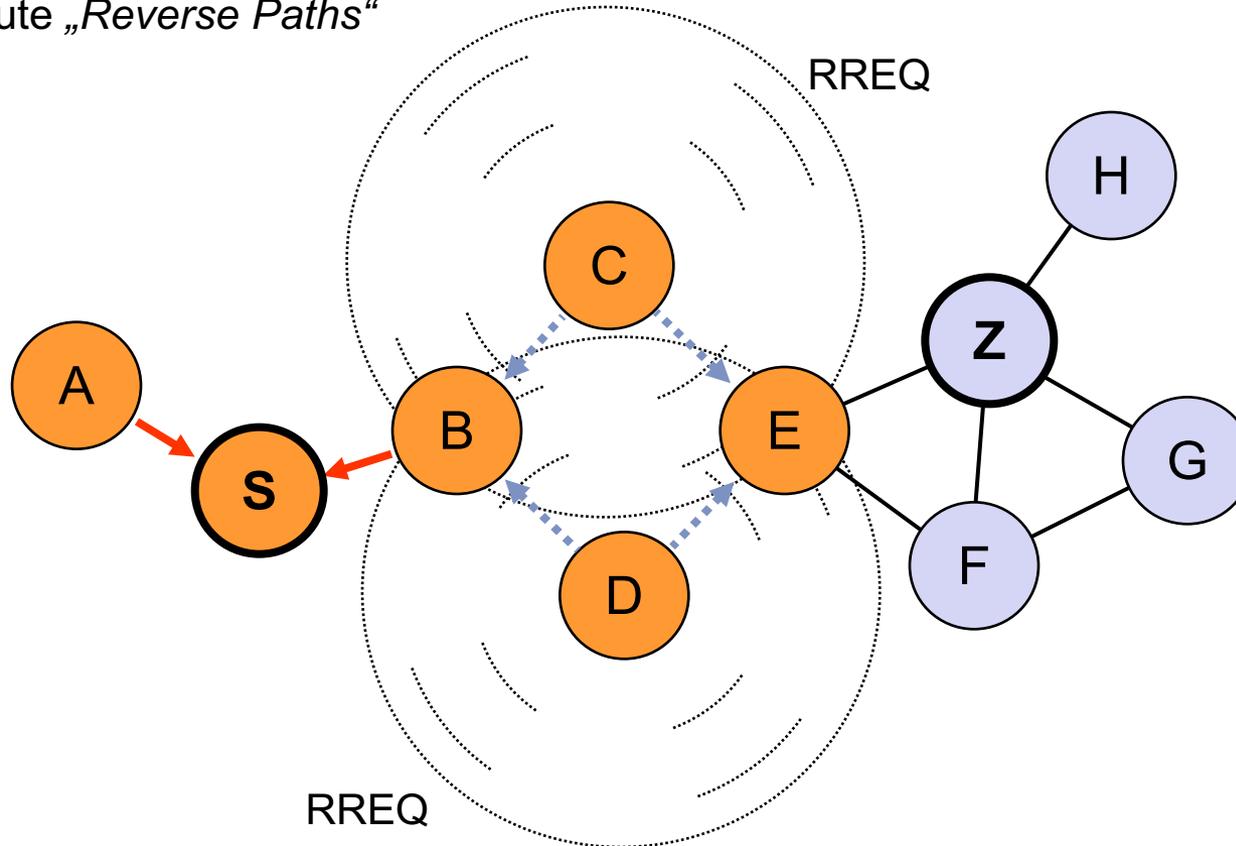
AODV: Fluten des RREQs III



- Jede Station, der das RREQ empfängt leitet es per Broadcast weiter
- S leitet sein selbst versendetes RREQ nicht erneut weiter!

AODV: Fluten des RREQs IV

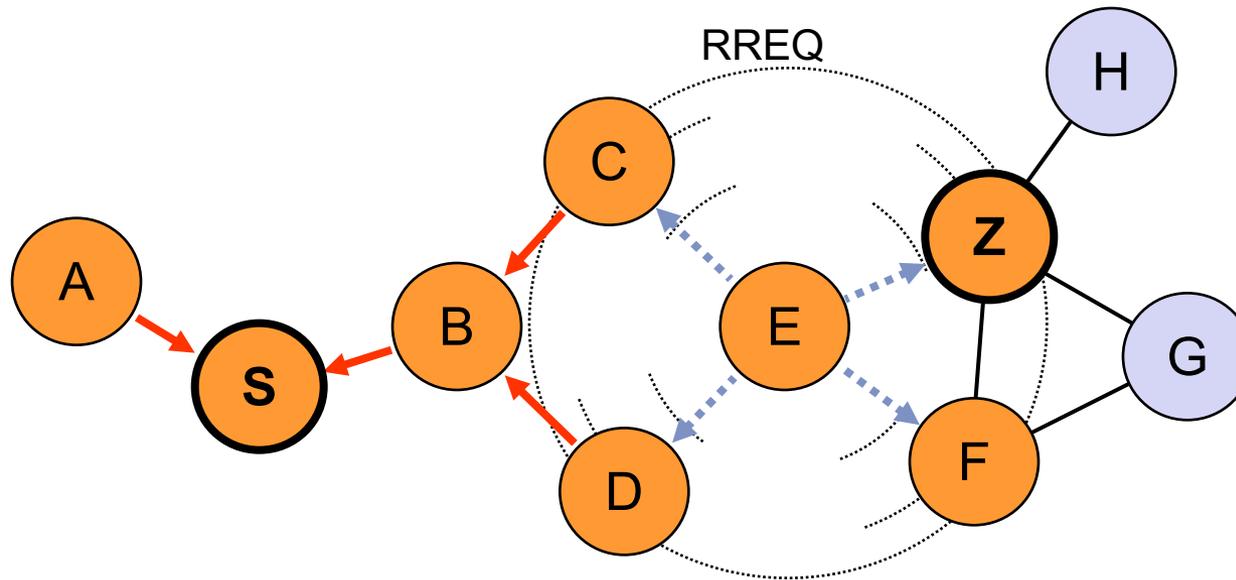
→ Aufgebaute „Reverse Paths“



- Beim Weiterleiten merken sich Stationen woher das RREQ empfangen wurde
 - Es wird die Station mit dem geringsten Hop-Count zu S gewählt
 - Es bildet sich der so genannte „reverse path“ zum Sender S

AODV: Fluten des RREQs V

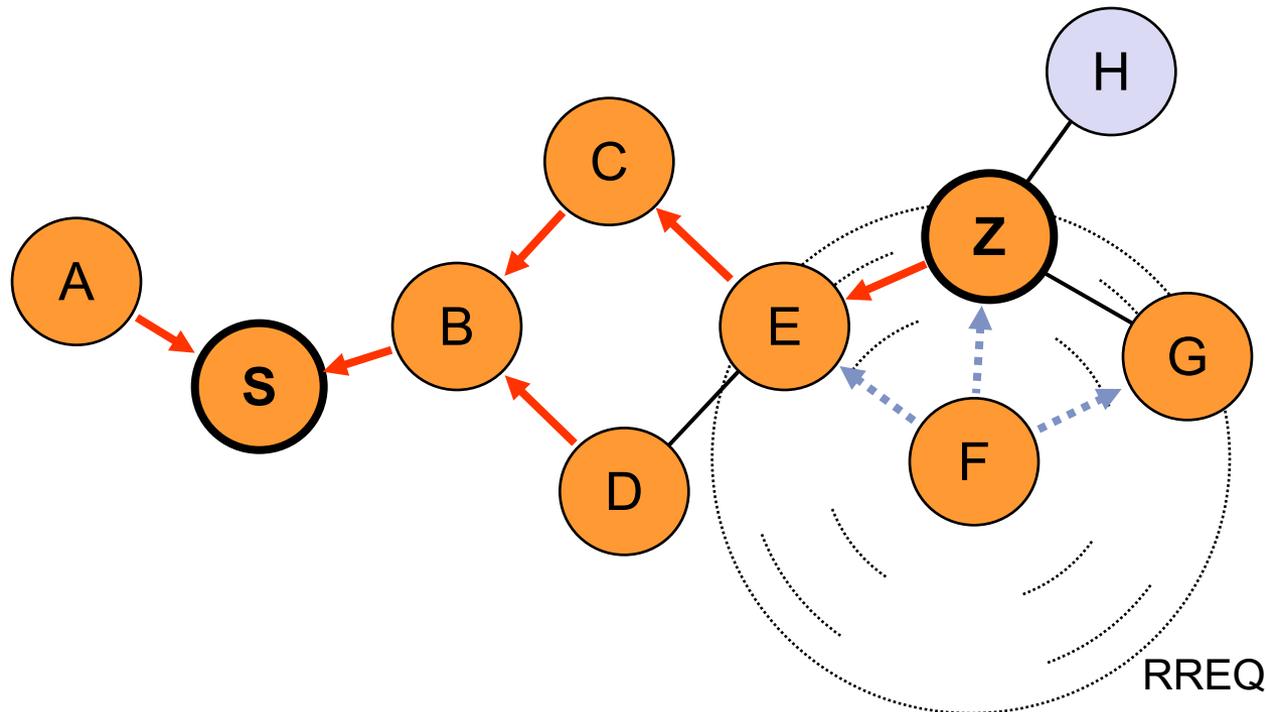
→ Aufgebaute „Reverse Paths“



- Duplikate werden durch Sequenznummern erkannt und nicht erneut weitergeleitet
 - Erfordert Zustandshaltung in den Stationen

AODV: Fluten des RREQs VI

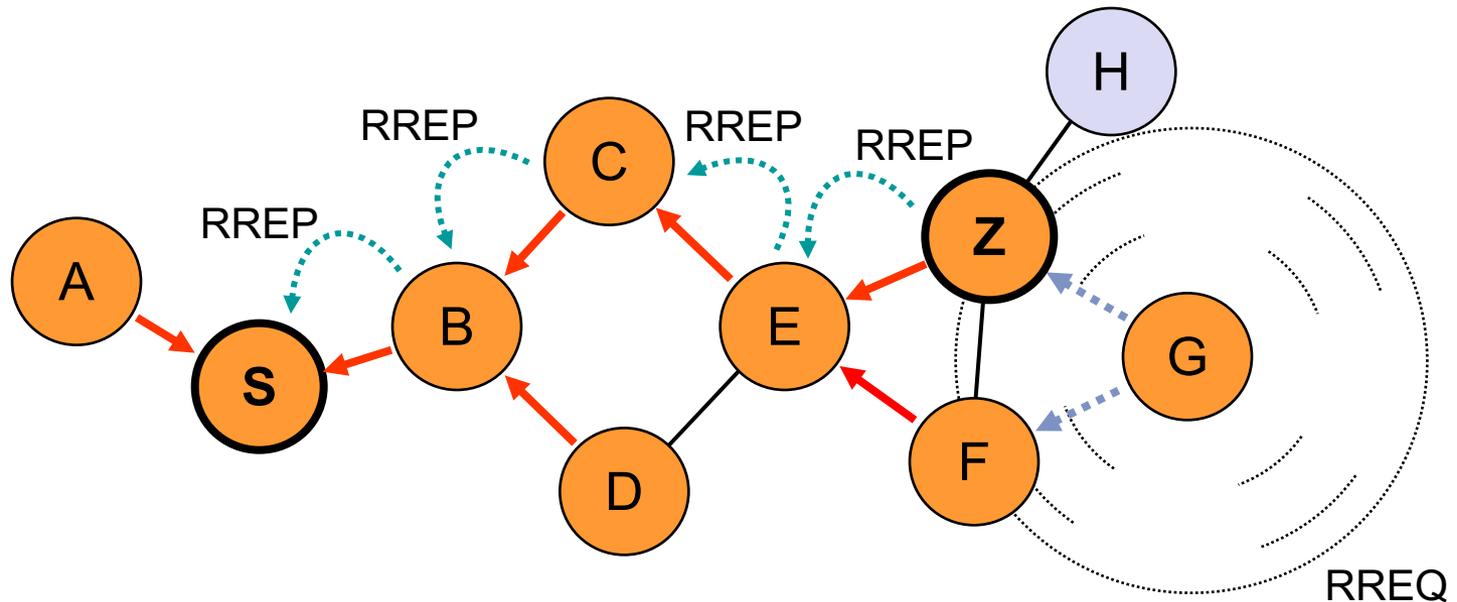
→ Aufgebaute „Reverse Paths“



- Das Ziel Z hat das von S ausgesendete RREQ empfangen
 - Es besteht jetzt ein *vollständiger „reverse path“* von Z zu S

AODV: Fluten des RREQs VII

→ Aufgebaute „Reverse Paths“

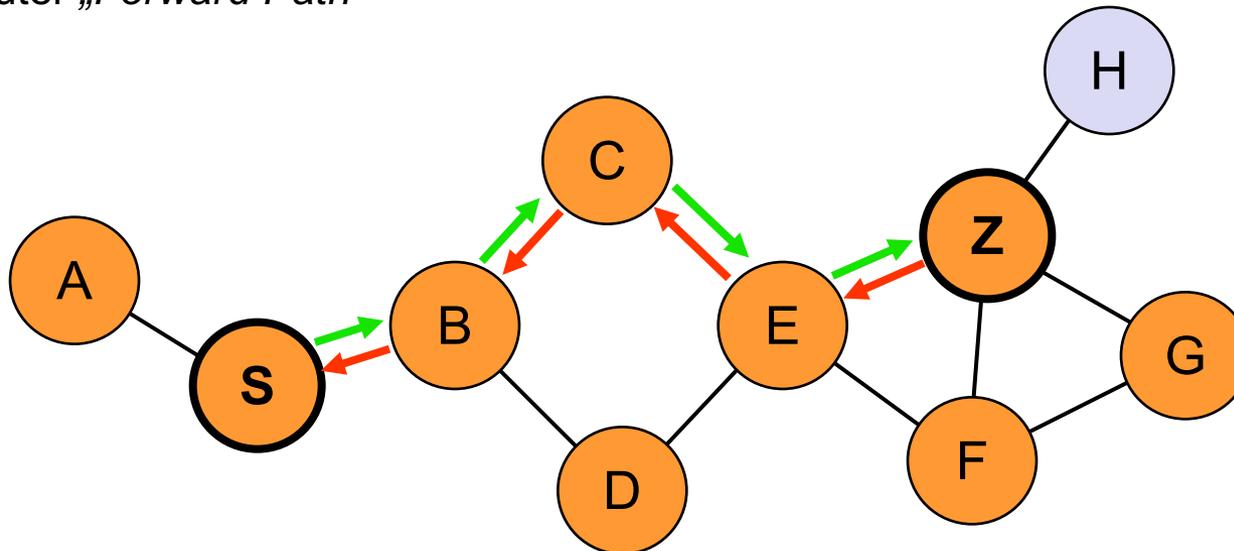


- Station Z sendet **RREP** über „reverse path“ zurück zu S
 - Next-Hop-Station auf „reverse path“ werden direkt adressiert
Übertragung per **Unicast**, keine Broadcasts mehr

AODV: Fluten des RREQs VII

→ Aufgebaute „Reverse Paths“

→ Aufgebauter „Forward Path“



- Es entsteht der „forward path“ von S zu Z
 - Dieser wird für den Datentransport von S zu Z verwendet
- Nicht verwendete „reverse path“-Einträge werden nach Timeout gelöscht
- Wurde Route von S zu Z aufgebaut, besteht auch Route von Z zu S

AODV: Sequenznummern

- Jede Station besitzt Zielsequenznummer (Destination Sequence Number)
 - Bestimmt die „Aktualität“ von Routinginformationen
 - Wird immer nur erhöht
 - Ausnahme: Überläufe

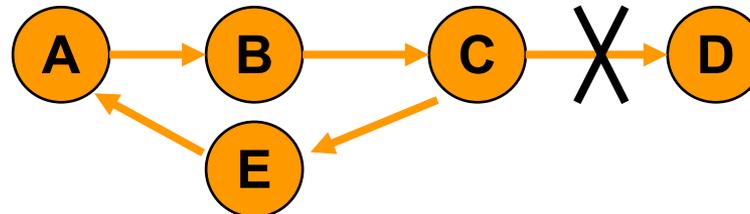
- RREQ enthält letzte bekannte Zielsequenznummer der Zielstation
 - Zielstationen oder *Zwischenstationen*, die eine neuere Route (höhere Zielsequenznummer) kennen, antworten mit RREP
 - RREP enthält ebenfalls Zielsequenznummer
 - Kann Aufbau von Routen beschleunigen
 - Wenn mehrere RREP empfangen werden, dann wird der RREP mit der höchsten Zielsequenznummer verwendet
 - Bei gleicher Zielsequenznummer entscheidet geringerer Hop Count

AODV: Aufrechterhalten einer Route

- Reverse/Forwarding Path Einträge werden nach Timeout gelöscht
 - Soft-state Ansatz
- Das Übertragen von Daten entlang einer Route frischt diese auf
 - Timer werden zurückgesetzt
 - Optional werden auch HELLO-Nachrichten zur Auffrischung eingesetzt
- Erkennung von Link-Brüchen
 - Fehlende HELLO-Nachrichten
 - Fehlende Link-layer Acknowledgements (Quittungen auf MAC-Schicht)
- Reaktion auf Link-Brüche: Versenden eines RERR
 - Wird über alle Zwischenstationen bis zu Quelle weitergeleitet
 - Quelle sucht neue Route mit einem RREQ
 - Enthält erhöhte Zielsequenznummer

AODV: Schleifenvermeidung

- Zielsequenznummern verhindern Schleifen
 - A sendet an D, Link von C nach D bricht
 - RERR von C nach A geht verloren
 - Später sucht C nach D
 - RREQ enthält höhere Zielsequenznummer
 - A empfängt RREQ über C-E-A
 - A sieht höhere Zielsequenznummer und lässt RREP aus



AODV: Optimierungen

■ Expanding Ring Search

- RREQs werden nicht im kompletten Netz geflutet
 - RREQ zunächst mit kurzer Lebenszeit (TTL) versehen
 - Bleibt Suche erfolglos, wird RREQ mit höherer TTL versendet
- Entlastet Netz bei Routenbestimmung zu nahe gelegenen Stationen
- Bei entfernten Stationen
 - Verzögert Routenaufbau
 - Nahe gelegene Stationen müssen viele RREQs weiterleiten

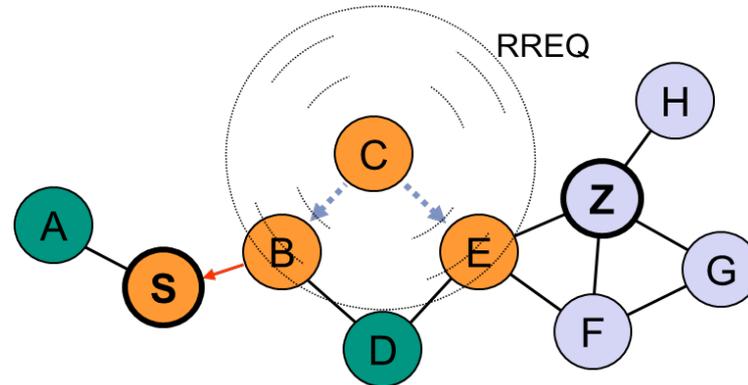
■ Local Route Repair

- Bei Link-Abbruch wird nicht sofort RERR zur Quelle gesendet
 - Station, die Link-Abbruch feststellt, sucht neue Route zum Ziel mittels RREQ
- Kann das Netz durchschnittlich entlasten
- Kann zu schlechteren (längeren) Routen führen, da nur Teilstrecken neu bestimmt werden

→ AODV ist „Standard-Protokoll“ (experimentell) in der IETF (RFC 3561)

DYMO und LOADng

- Aktuelle Weiterentwicklungen von AODV
- **Dynamic MANET On-demand (DYMO, aka AODVv2) Routing**
 - RREQ beinhaltet **Liste aller weiterleitenden Stationen**
 - Stationen, die RREQ weiterleiten können zusätzliche Routen in Tabelle aufnehmen
 - Bsp: C leitet RREQ (von S über B) an E weiter
 - E kennt jetzt Routen zu S, B und C

 [10.5]


- **Lightweight On-demand Ad hoc Distance-vector Routing Protocol – Next Generation (LOADng)**
 - Nur Ziel antwortet auf RREQ
 - RRER nur an Quelle einer Nachricht
 - Unterstützt optimiertes Fluten (s.u.) und verschiedene Adresslängen
 - Großes industrielles Interesse

 [10.7]

- Ziel: Link Status Information *effizient* fluten!

1. Bestimmung der Nachbarschaft

- Periodisches broadcasten von HELLO-Nachrichten. Diese enthalten:
 - Adresse der Stationen, von denen derzeit HELLO-Nachrichten empfangen werden
 - Status der Links zu diesen Stationen: symmetrisch (bi-) oder asymmetrisch (unidirektional)
- Daraus errechenbare Nachbarschaftsbeziehungen
 - Y ist „1-hop Nachbar“ von X \Leftrightarrow X empfängt HELLO-Nachricht von Y
 - Y ist „2-hop Nachbar“ von X \Leftrightarrow Y \neq X und X sieht Y in von Z empfangener HELLO-Nachricht
 - Y ist „striker 2-hop Nachbar“ von X \Leftrightarrow Y ist 2-hop aber nicht 1-hop Nachbar von X
 - Beziehungen sind jeweils symmetrisch und asymmetrisch definiert
 - Bsp.: X sieht sich in HELLO-Nachricht von Y \Rightarrow Y ist symmetrischer 1-hop Nachbar von X
- Nachbarschaftsinformation nach Timeout löschen zur Reaktion auf Link-Brüche



[10.6]

Multipoint Relays (MPR)

2. Bestimmung der „Multipoint Relays“ (MPR)

- Die MPRs von X sind eine Teilmenge der symmetrischen 1-hop Nachbarn von X
- Die Teilmenge wird so gewählt, dass jeder symmetrische 2-hop Nachbar von X über mindestens einen MPR erreichbar ist
- Berechnung der Teilmenge anhand einer Heuristik bei jeder erkannten Änderung innerhalb der 2-hop Nachbarschaft
- Gewählte MPRs werden in HELLO-Nachrichten bekannt gegeben

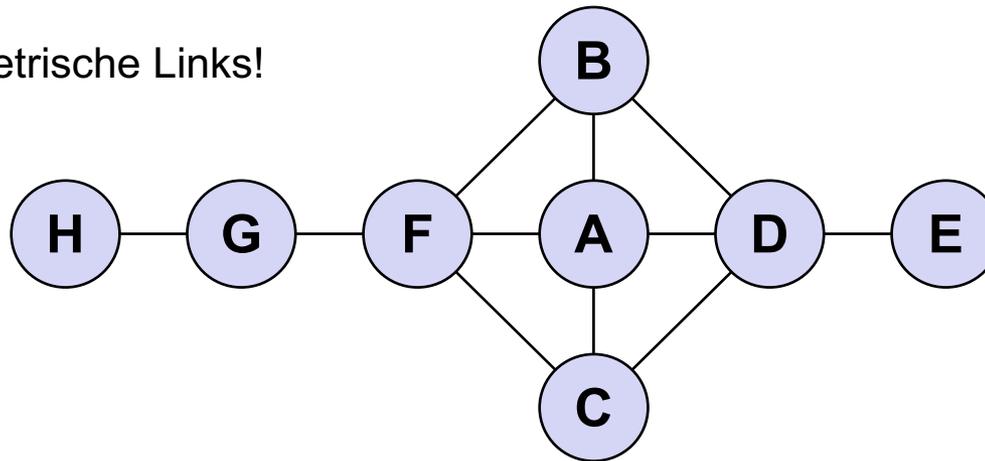
3. Bestimmung der „MPR Selectors“ (MS)

- Die MS von X sind jene Stationen, welche X als MPR gewählt haben
- X erfährt von diesen Stationen aufgrund empfangener HELLO-Nachrichten

Beispiel OLSR I

Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)

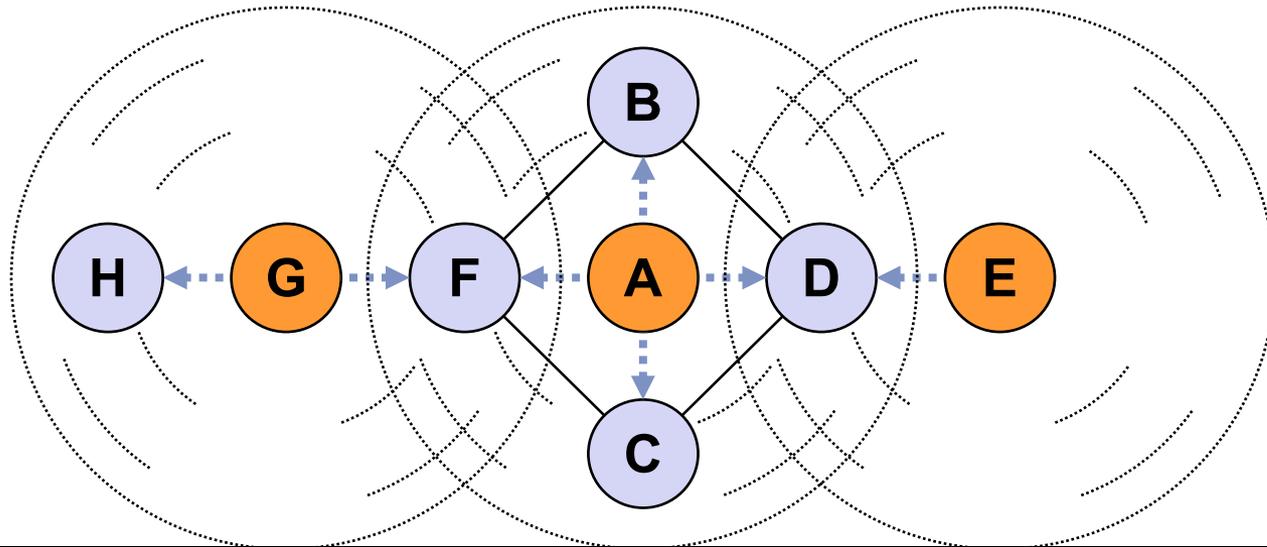
Hier nur symmetrische Links!



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A				
B				
C				
D				
E				
F				
G				
H				

Beispiel OLSR II

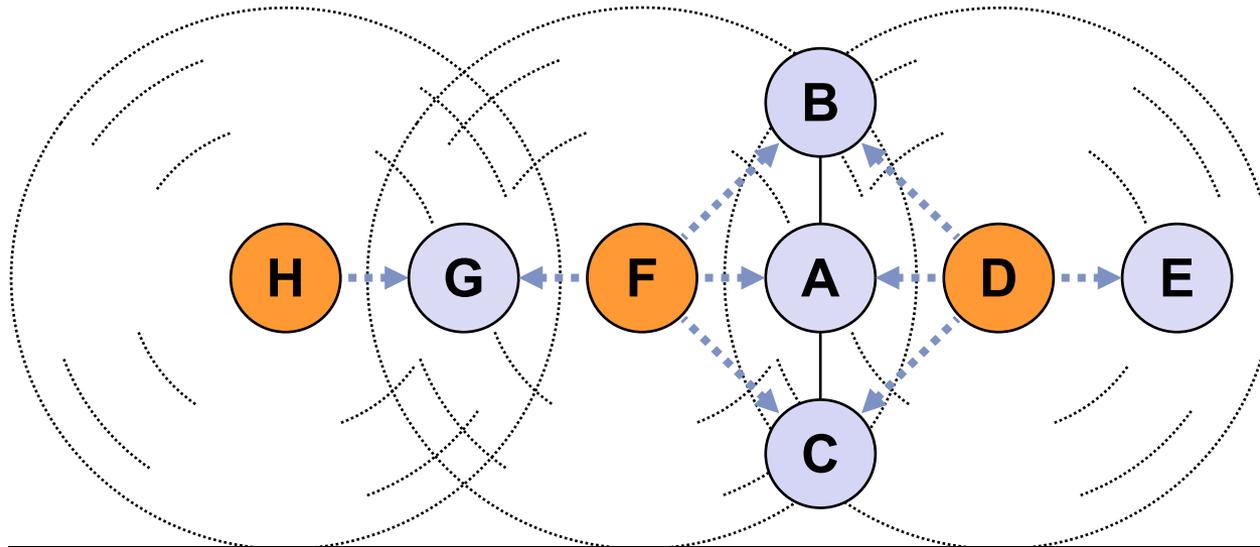
Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A				
B	A			
C	A			
D	A, E			
E				
F	A, G			
G				
H	G			

Beispiel OLSR III

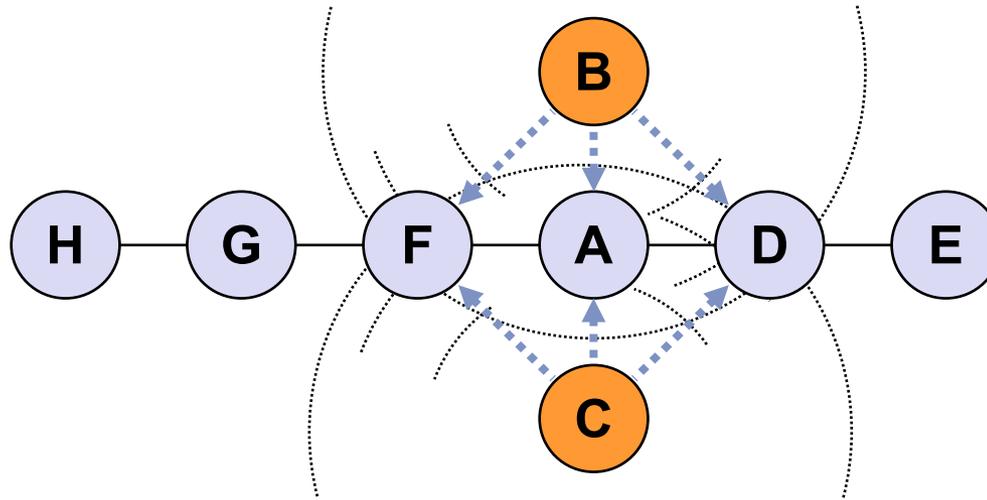
Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A	F, D	E, G	D, F	
B	A, D, F	E, G	D, F	
C	A, D, F	E, G	D, F	
D	A, E			
E	D	A	D	
F	A, G			
G	F, H	A	F	
H	G			

Beispiel OLSR IV

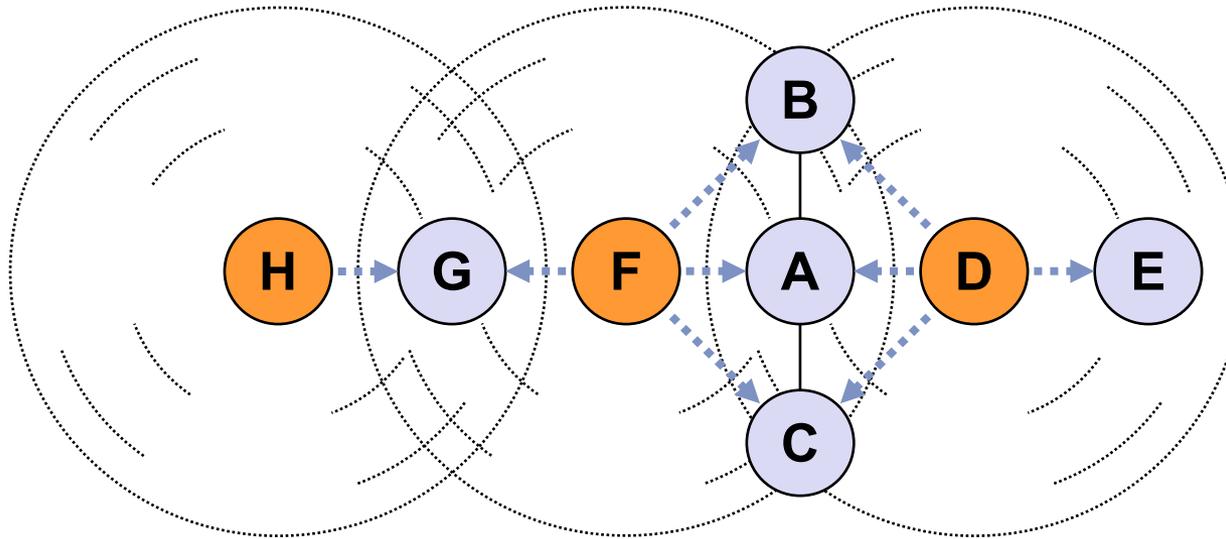
Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A	B, C, D, F	E, G	D, F	
B	A, D, F	E, G	D, F	
C	A, D, F	E, G	D, F	
D	A, B, C, E	F	B	B, C
E	D	A	D	
F	A, B, C, G	D	B	B, C
G	F, H	A	F	
H	G			

Beispiel OLSR V

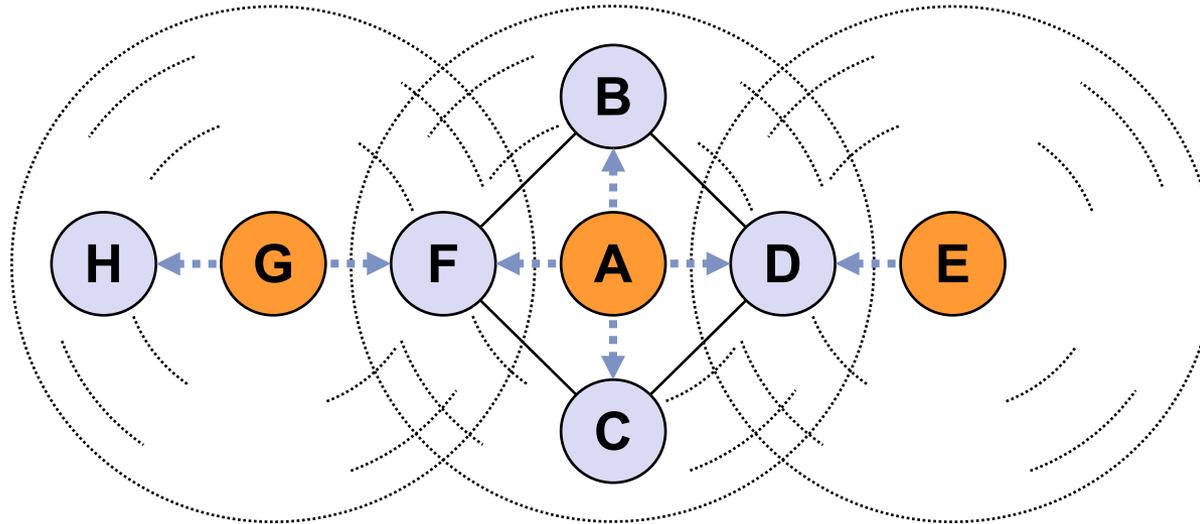
Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A	B, C, D, F	E, G	D, F	
B	A, D, F	C, E, G	D, F	D, F
C	A, D, F	B, E, G	D, F	
D	A, B, C, E	F	B	B, C
E	D	A, B, C	D	
F	A, B, C, G	D	B	B, C
G	F, H	A, B, C	F	
H	G			

Beispiel OLSR VI

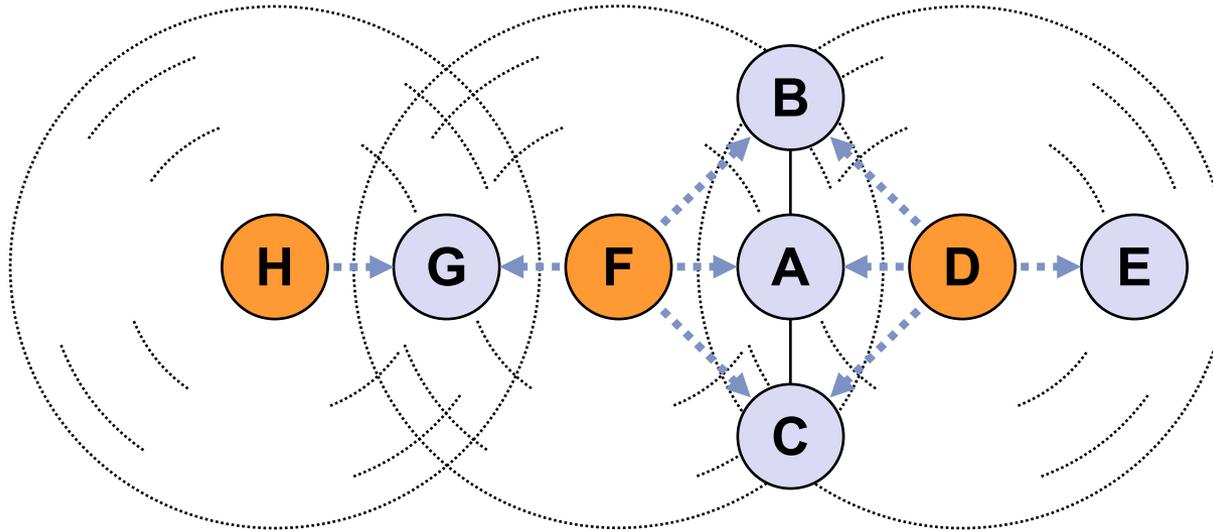
Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A	B, C, D, F	E, G	D, F	
B	A, D, F	C, E, G	D, F	D, F
C	A, D, F	B, E, G	D, F	
D	A, B, C, E	F	B	A, B, C, E
E	D	A, B, C	D	
F	A, B, C, G	D, H	B, G	A, B, C, G
G	F, H	A, B, C	F	
H	G	F	G	

Beispiel OLSR VII

Bestimmung der Nachbarn, MPR und MS (HELLO-Nachrichten)



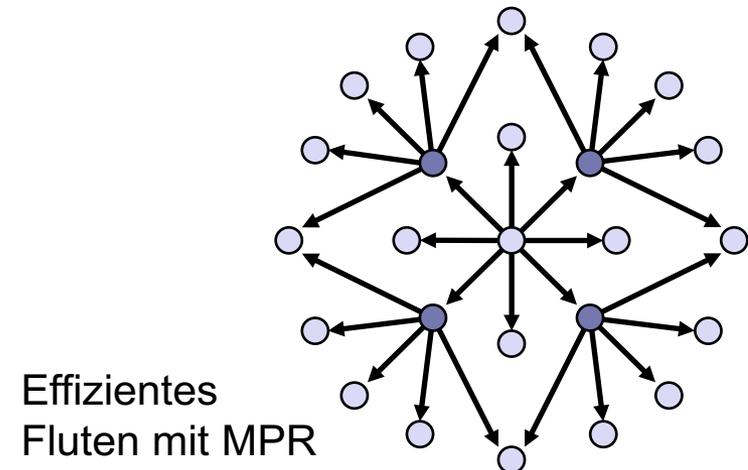
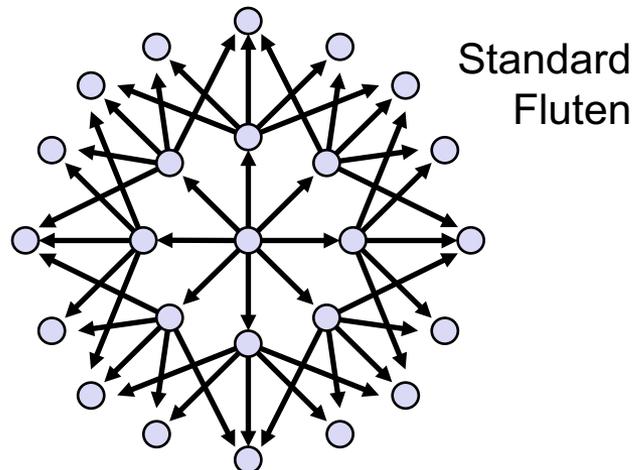
**S
T
A
B
I
L
!**

Station	1-hop Nachbarn	2-hop Nachbarn	MPR	MS
A	B, C, D, F	E, G	D, F	
B	A, D, F	C, E, G	D, F	D, F
C	A, D, F	B, E, G	D, F	
D	A, B, C, E	F	B	A, B, C, E
E	D	A, B, C	D	
F	A, B, C, G	D, H	B, G	A, B, C, G
G	F, H	A, B, C	F	F, H
H	G	F	G	

**S
T
A
B
I
L
!**

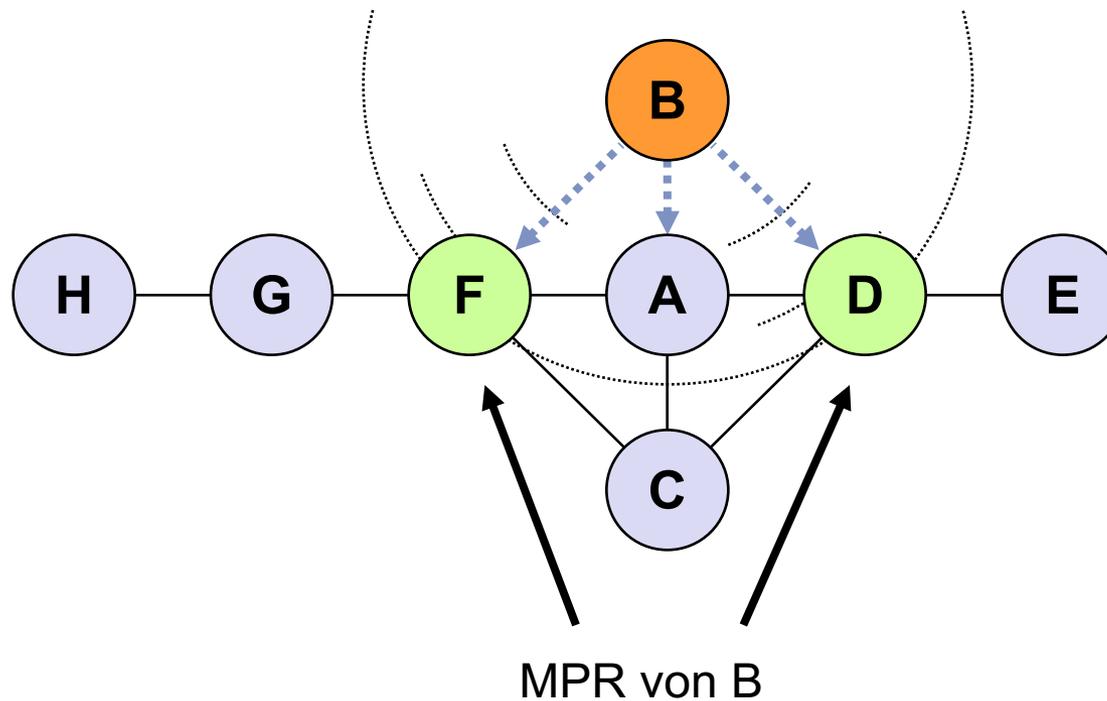
Optimized Link State Routing (OLSR)

- Hiermit möglich: Effizientes Fluten von Nachrichten
 - Quellstationen broadcastet Nachricht
 - Optimierung gegenüber dem Standard Fluten-Algorithmus:
 - Nicht *alle* Stationen leiten Nachricht weiter, sondern *nur* die MPR
 - Y leitet eine von X empfangene Nachricht weiter, falls X ein MS von Y ist
 - Weiterleitung ebenfalls als Broadcast
 - Vermeidung der Weiterleitung von Duplikaten durch Sequenznummern
 - Zahl der gewählten MPR wirkt sich direkt auf die Netzbelastung aus
 - Menge der MPR sollte möglichst optimal (minimal) sein, um Verkehr zu reduzieren



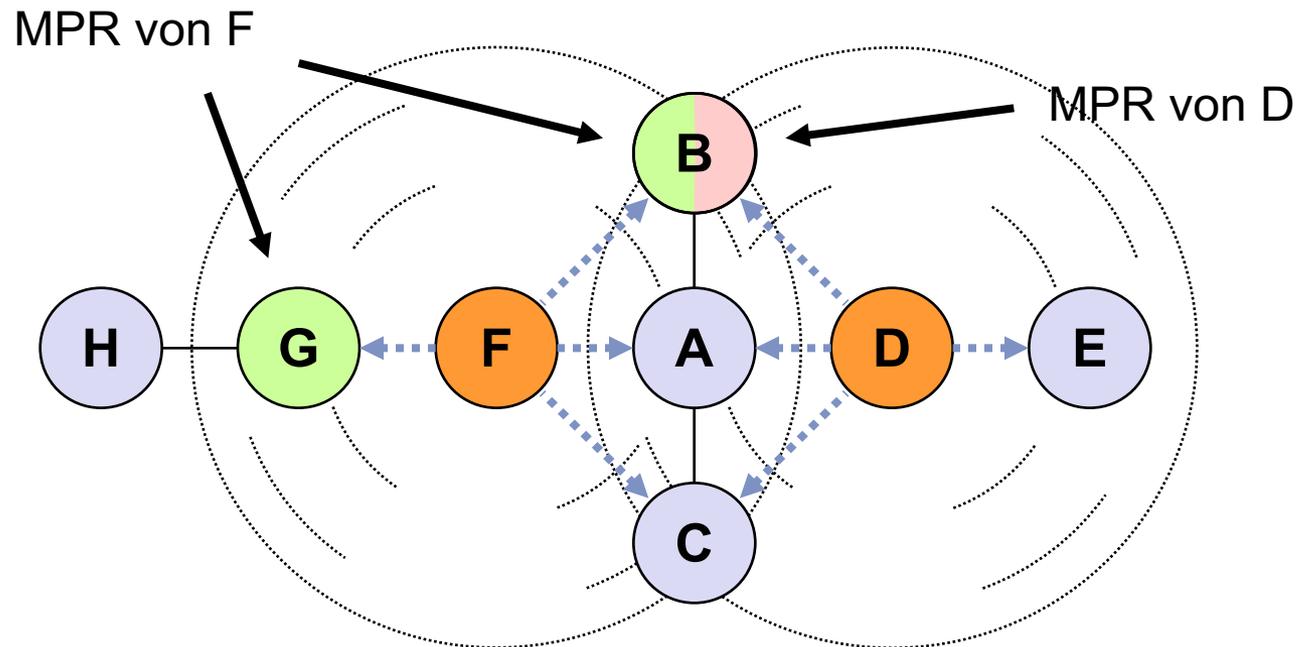
Beispiel OLSR VIII

Effizientes Fluten von Nachrichten, Beispiel: Quellstation B



Beispiel OLSR IX

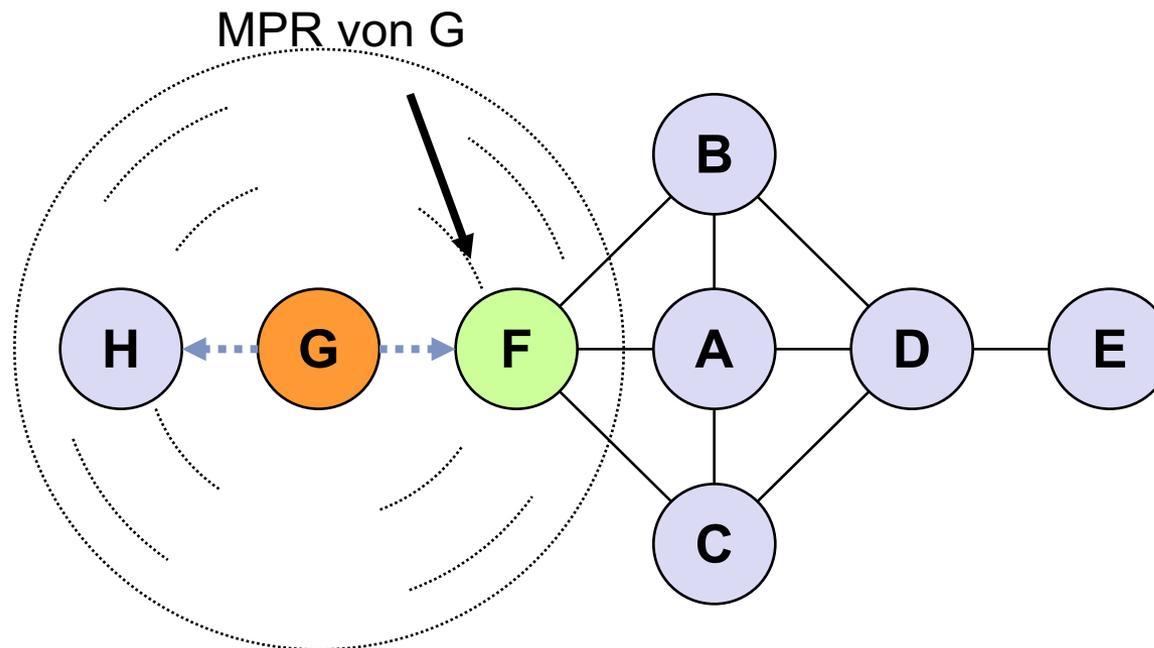
Effizientes Fluten von Nachrichten, Beispiel: Quellstation B



- Stationen F und D sind MPR von B und leiten die empfangene Nachricht weiter

Beispiel OLSR X

Effizientes Fluten von Nachrichten, Beispiel: Quellstation B



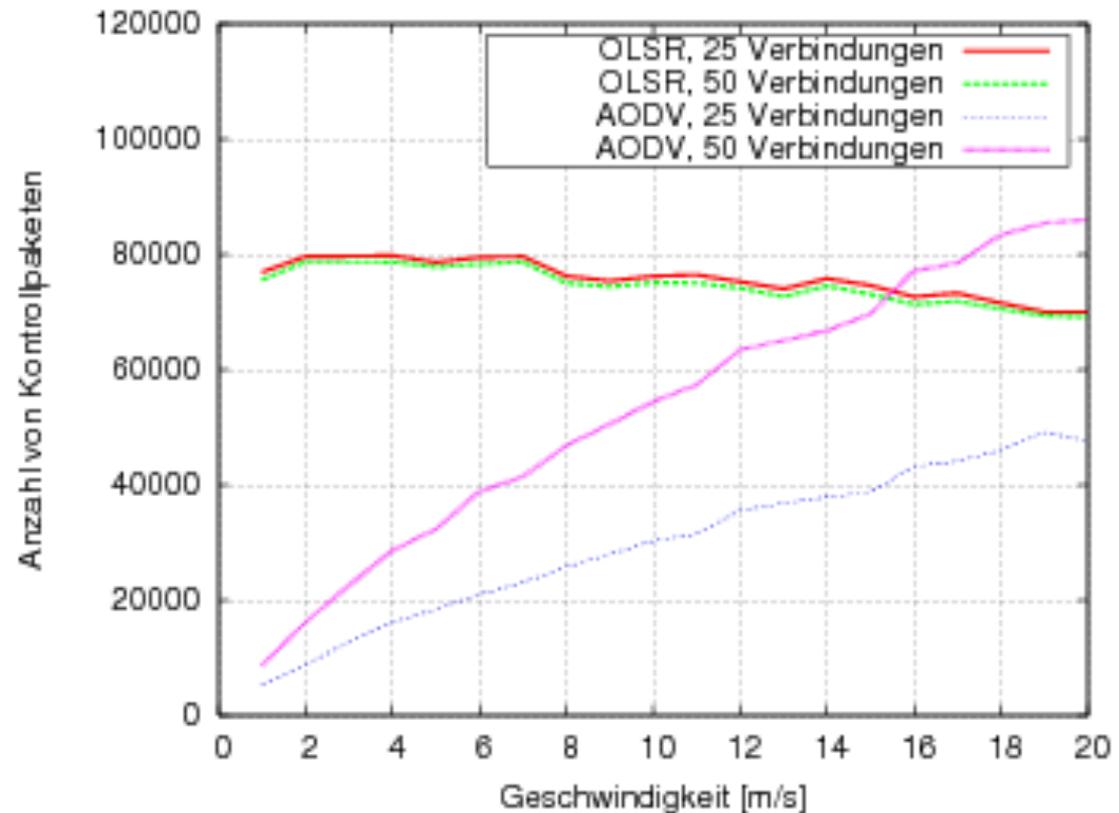
- Station B leitet die von F bzw. D empfangene Nachricht nicht erneut weiter (Sequenznummern)
- Station G ist MPR von F und leitet die von F empfangene Nachricht weiter
- Station F leitet die von G empfangene Nachricht nicht erneut weiter (Sequenznr.)

Optimized Link State Routing (OLSR)

- Effizientes Fluten von Link Status Informationen
 - Jede Station X mit $MS \neq \emptyset$ versendet periodisch **Topology Control (TC)** Nachricht
 - Enthält eigene Adresse
 - Komplette Liste der eigenen MS
 - Es werden also *nicht* die 1-hop Nachbarn propagiert! Grund:
 - Es sollen nur symmetrische Links propagiert werden
 - MS bilden *Teilmenge* der Stationen, zu denen symmetrische Links bestehen
 - Reduktion der Netzbelastung
 - Nicht alle Stationen senden TC-Nachrichten
 - Jede Station im Netz sammelt MS aus TC-Nachrichten
 - Netztopologie ist in ausreichendem Maß rekonstruierbar
 - Kürzeste Routen können nach Dijkstra ermittelt werden
 - Optimierung: Piggyback von HELLO- und TC-Nachrichten
 - Weniger Medienzugriffe
 - Besonders effektiv bei Netzen mit hoher Gerätedichte
 - Nur wenige Stationen fungieren als MPR
- OLSR ist „Standard-Protokoll“ (experimentell) in der IETF (RFC 3626)

Vergleich AODV – OLSR

Anzahl Stationen	50
Simulationsfläche	1000m x 1000m
Simulationsdauer	600 Sekunden
Reichweite	250 m
Funktechnologie	IEEE 802.11
Mobilitätsmodell	Random Waypoint



→ Beobachtungen

- Kontrolloverhead von AODV steigt mit zunehmender Geschwindigkeit
- Kontrolloverhead von AODV steigt mit der Anzahl der Verbindungen
- Kontrolloverhead von OLSR ist unabhängig von der Anzahl der Verbindungen
- Kontrolloverhead von OLSR ist nahezu unabhängig von der Geschwindigkeit

Zusammenfassung

- Mobile Ad Hoc Netze (MANET) ermöglichen Aufbau komplexer Netztopologien ohne Infrastruktur-Unterstützung
 - Mobilstationen fungieren selber als Router

- Routing-Protokolle aus Festnetz nicht für MANET geeignet

- Kategorisierung MANET Routing-Protokolle
 - Proaktiv
 - Reaktiv
 - Hybrid

- Eignung eines Protokoll hängt stark vom Szenario ab!

- 10.1 Was sind mobile Ad Hoc Netze und was sind deren Eigenschaften?
- 10.2 Was ist der Unterschied zwischen mobilen Ad Hoc Netzen und IEEE 802.11 im Ad Hoc Modus?
- 10.3 Warum können aus dem Festnetz bekannte Protokolle nicht in mobilen Ad Hoc Netzen angewandt werden?
- 10.4 Benennen und beschreiben Sie die besprochenen Kategorien von Routingprotokollen und nennen Sie je ein Beispielprotokoll!
- 10.5 Was sind Vor- und Nachteile des Flutens für den Datentransfer?
- 10.6 Wie funktioniert AODV?
- 10.7 Was ist die Besonderheit bei OLSR?
- 10.8 In welcher Situation würden Sie welches Protokoll anwenden?

Referenzen und weiterführende Literatur

- [10.1] The MANET Bibliography
http://www.antd.nist.gov/wctg/manet/manet_bibliog.html
- [10.2] IETF MANET Working Group <http://www.ietf.org/html.charters/manet-charter.html>
- [10.3] C. Perkins, Ad-hoc Networking, Addison Wesley, 2000
- [10.4] C. Perkins et al., Ad hoc On-Demand Distance Vector (AODV) Routing, IETF RFC 3561, <http://www.ietf.org/rfc/rfc3561.txt>
- [10.5] C. Perkins et al., Dynamic MANET On-demand (AODVv2) Routing, IETF Internet Draft, January 2013, <http://tools.ietf.org/html/draft-ietf-manet-dymo-25> (work in progress)
- [10.6] T. Clausen et al., Optimized Link State Routing Protocol (OLSR),
<http://www.ietf.org/rfc/rfc3626.txt>
- [10.7] T. Clausen et al., The Lightweight On-demand Ad hoc Distance-vector Routing Protocol – Next Generation (LOADng), IETF Internet Draft, January 2013,
<http://tools.ietf.org/html/draft-clausen-ltn-loadng-08> (work in progress)