

Selbstorganisierende Netze

Routing-Protokolle und technische Aspekte

Agenda

- Einleitung
 - Motivation
 - aktuelle Marktsituation
- Routing-Protokolle
 - Besondere Anforderungen
 - Adressierungssysteme
 - Routing-Ansätze und Klassifizierung von Protokollen
- Datenintegrität, Sicherheit und Privatsphäre
- Noch zu lösende Problemfelder

Motivation

- „Wozu Mesh-Netzwerke“?
 - Anwendungsgebiete finden sich überall dort, wo klassische Funknetze unwirtschaftlich oder zu unflexibel sind
 - Veranstaltungen wie z.B. Messen oder Kongresse
 - Entwicklungs- und Schwellenländer
 - Verkehrsvernetzung, Car-to-Car-Communication
 - Sensor-Netzwerke über größere Gebiete

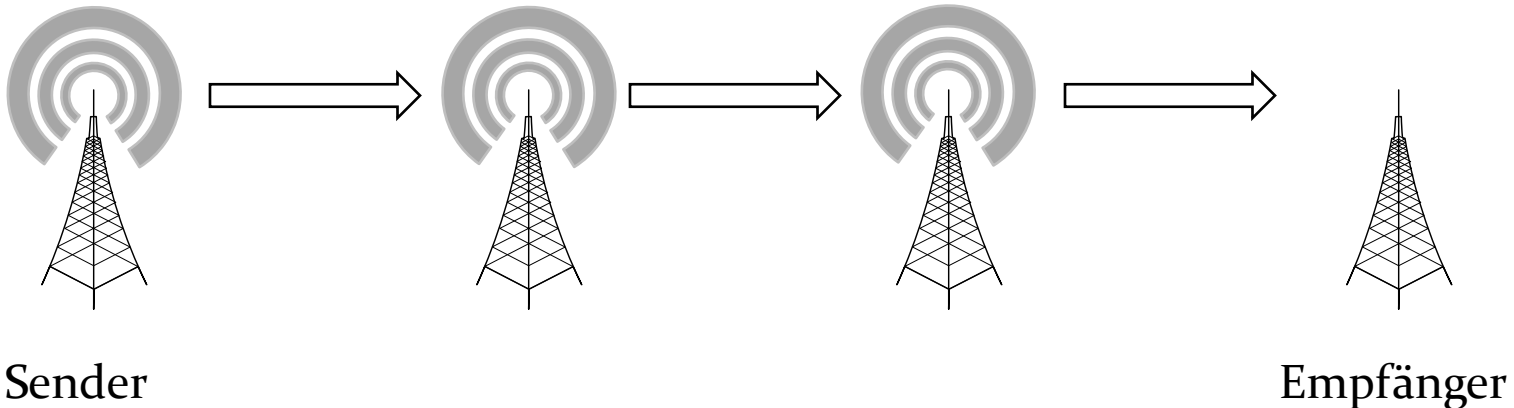
Aktuelle Marktsituation

- Derzeit existieren nur sehr wenige in größerem Stil eingesetzte Mesh-Netze
 - Das OLPC-Projekt setzt Mesh-Netzwerktechnik zur Vernetzung der XO-Laptops ein
 - Im Bereich der Car2Car-Communication werden erste Pilotprojekte durchgeführt
 - Zum Aufbau drahtloser Sensornetze sind Lösungen verfügbar, die Mesh-Technik einsetzen



Anforderungen an Routing-Protokolle

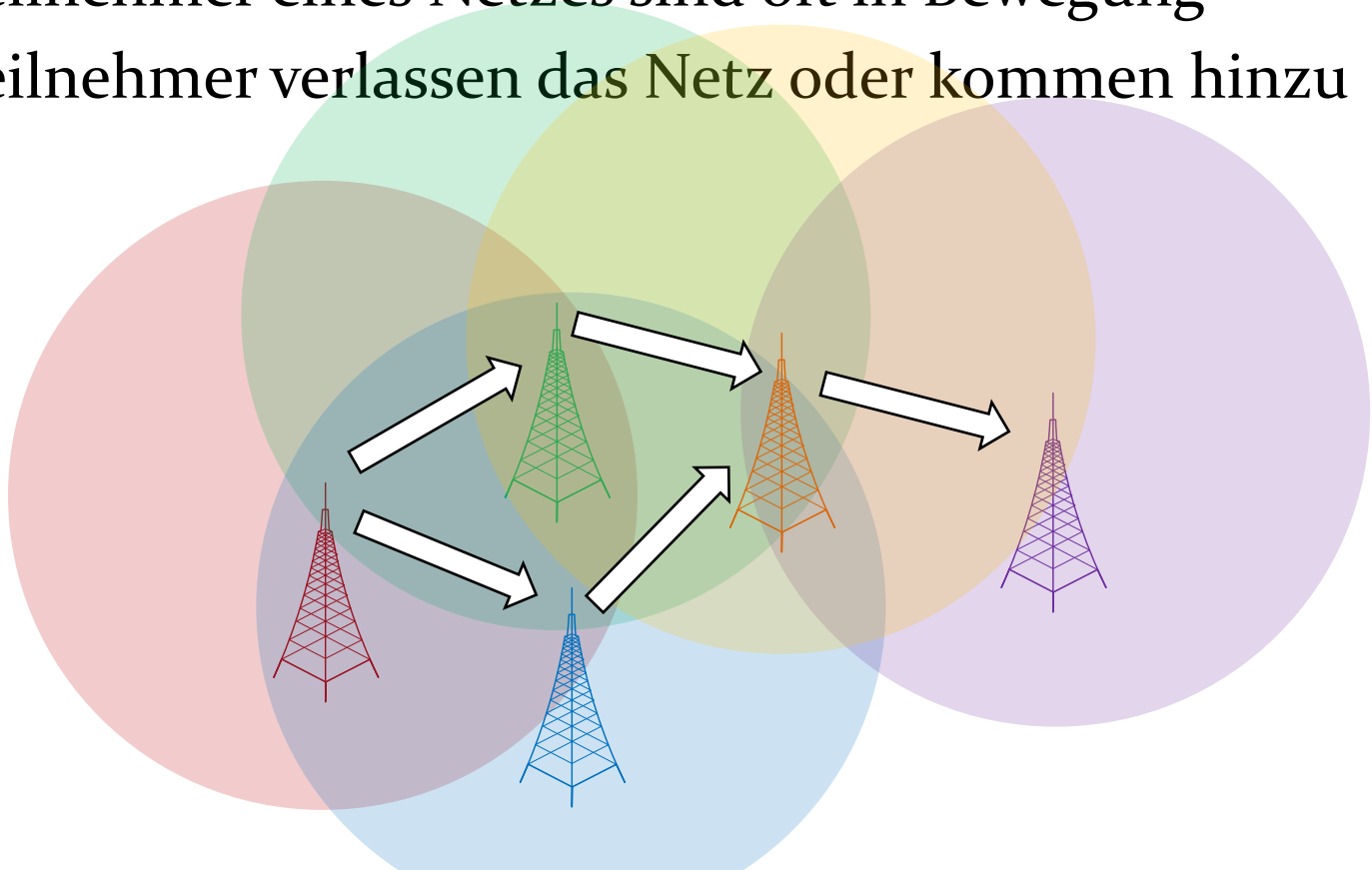
- Effiziente Bandbreitennutzung
 - Funkkanal -> begrenzte Bandbreite, „Shared Medium“
 - Oft mehrfache Belegung des Funkkanals



- Daher gefordert: möglichst wenig Protokoll-Overhead!

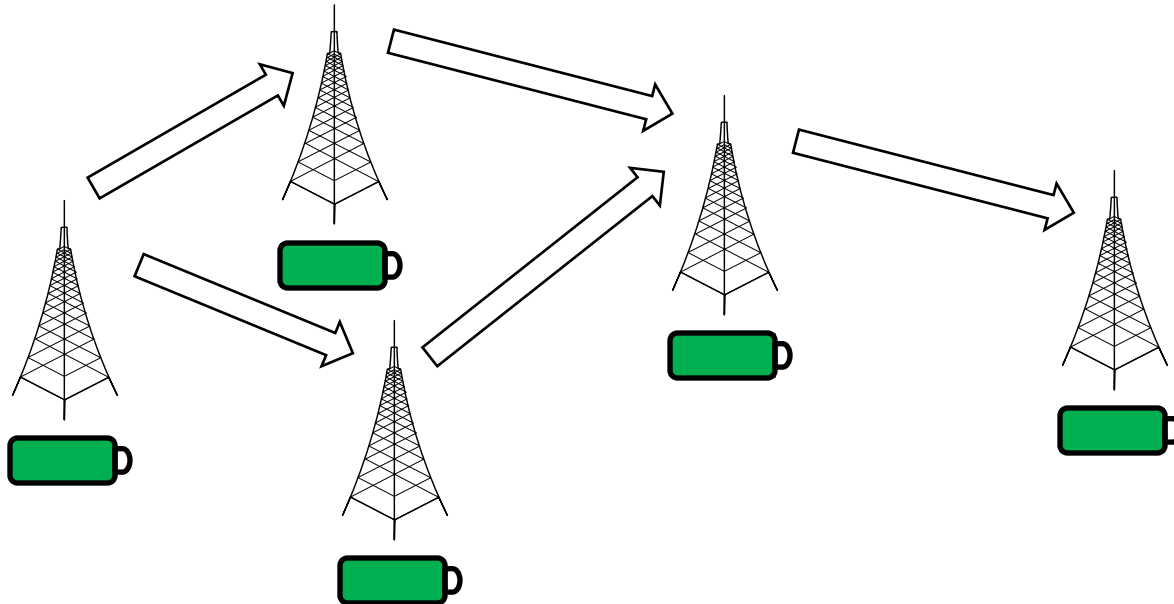
Anforderungen an Routing-Protokolle

- Stabilität bei häufigen Topologieänderungen
 - Teilnehmer eines Netzes sind oft in Bewegung
 - Teilnehmer verlassen das Netz oder kommen hinzu



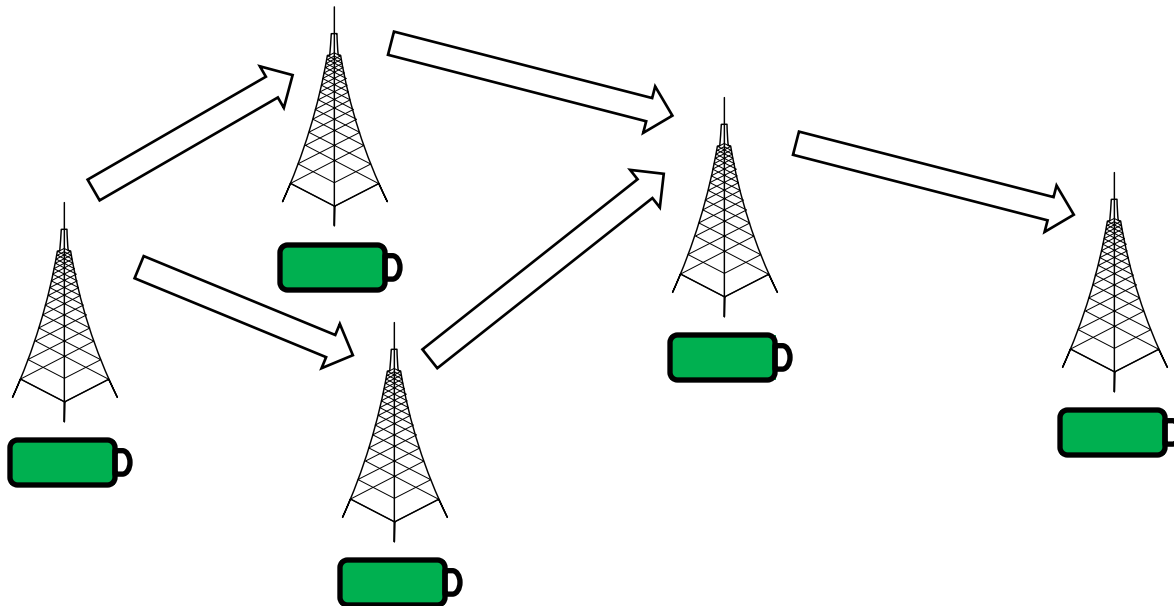
Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



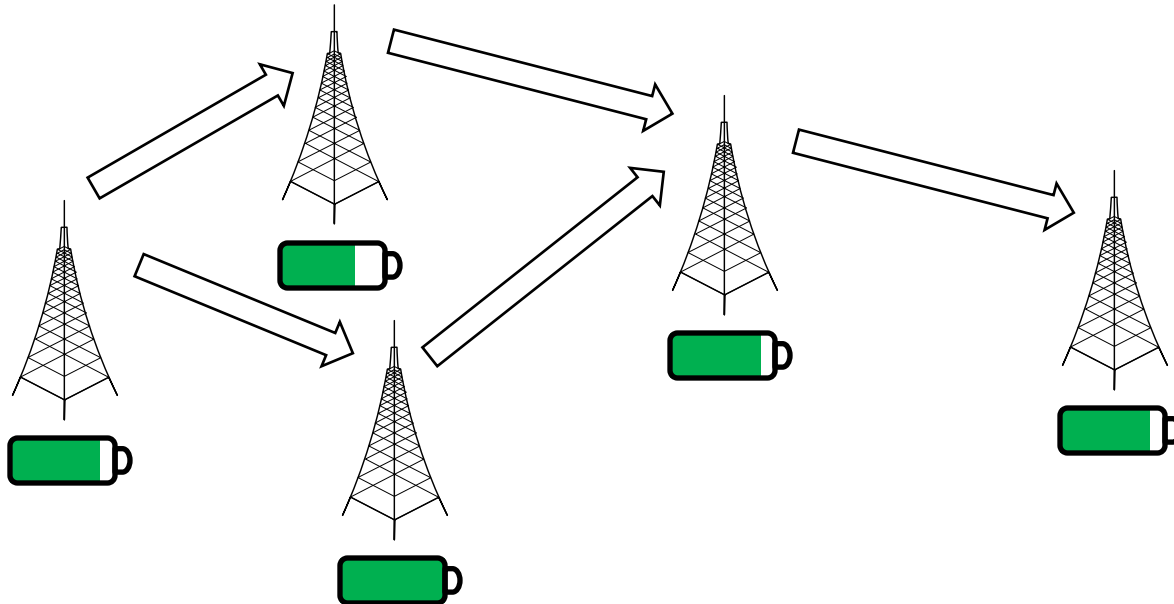
Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



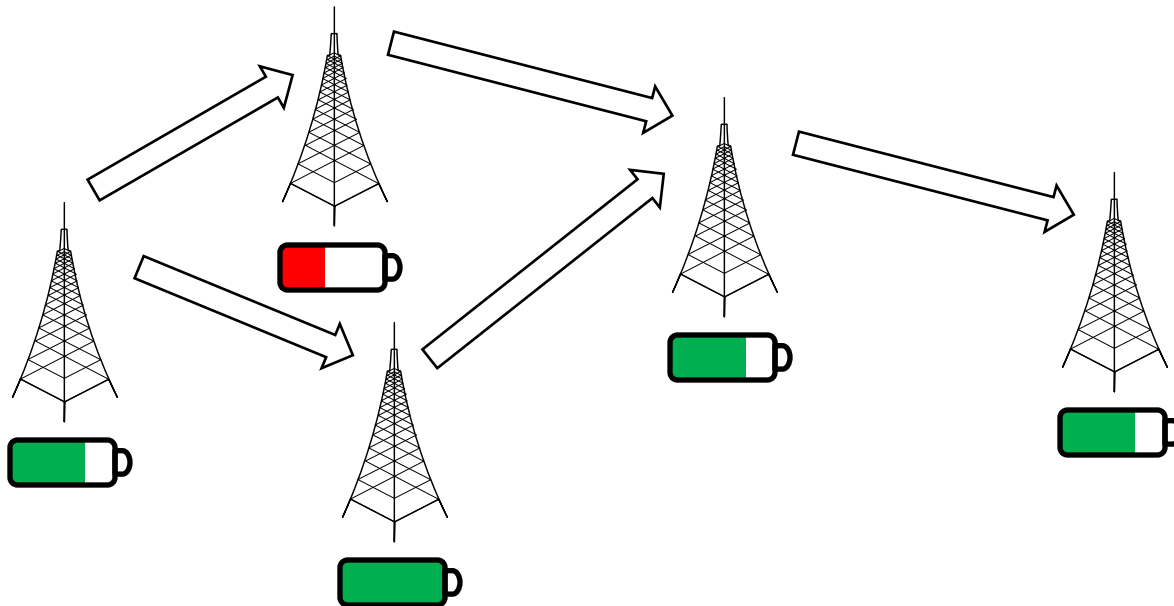
Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



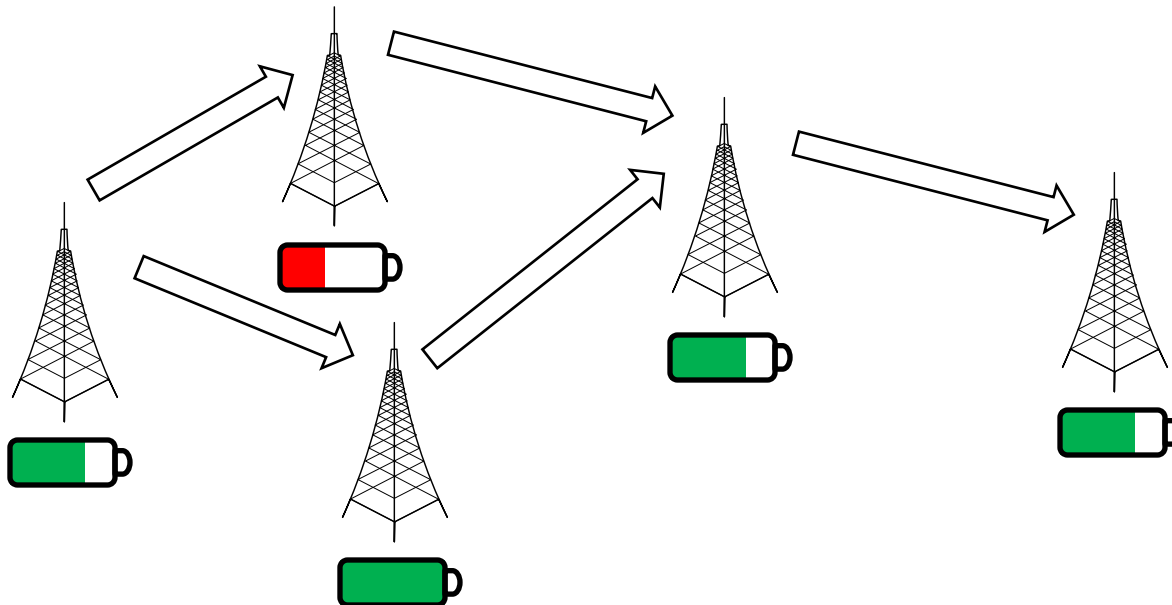
Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



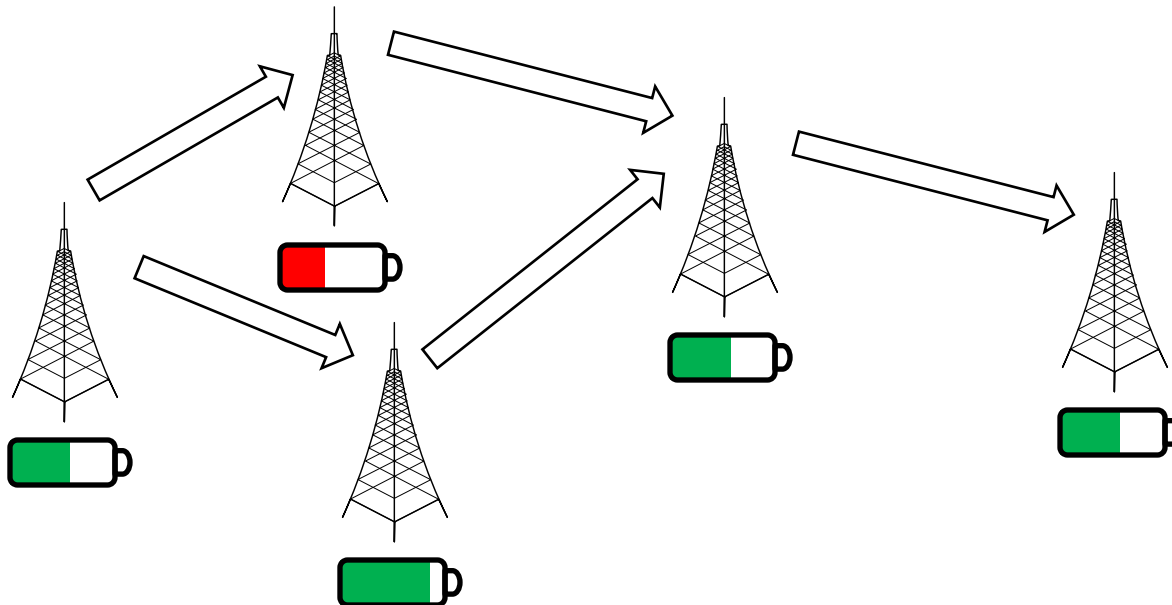
Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



Anforderungen an Routing-Protokolle

- Berücksichtigung von Umgebungseinflüssen
 - Umgebungseinflüsse können z.B. Interferenzen, Frequenzstörungen, aber auch die verfügbare Akkuleistung sein



Adressierungssysteme

- Adresse für jeden Knoten unabdingbar für Routing
- Adressvergabesysteme können nach Eindeutigkeit der Adressen klassifiziert werden
 - **Global eindeutig**
 - Jede Adresse existiert nur einmal weltweit
 - Oft realisiert mittels bei der Produktion vergebenen festen Adressen
 - Vorteil: Kollisionen sind (fast) ausgeschlossen, keine Adressvergabe im Netz selbst notwendig
 - Nachteil: Benötigt großen Adressraum -> erhöht Protokoll-Overhead, ermöglicht Tracking einzelner Knoten

Adressierungssysteme

- Adresse für jeden Knoten unabdingbar für Routing
- Adressvergabesysteme können nach Eindeutigkeit der Adressen klassifiziert werden
 - **Global eindeutig**
 - **Netzwerkweit eindeutig**
 - Jede Adresse existiert in einem Netzwerk nur einmal
 - Adresse kann mehrfach in verschiedenen Netzen vorkommen
 - Vorteil: Erheblich kleinerer Adressraum erforderlich, bei dyn. Adressvergabe kein Tracking eines Knotens möglich
 - Nachteil: Erfordert System zur Adressvergabe beim Einklinken in ein Netzwerk

Adressierungssysteme

- Adresse für jeden Knoten unabdingbar für Routing
- Adressvergabesysteme können nach Eindeutigkeit der Adressen klassifiziert werden
 - **Global eindeutig**
 - **Netzwerkweit eindeutig**
 - **Lokal eindeutig**
 - Adressen sind innerhalb einer „Nachbarschaft“ eindeutig
 - Vorteil: nochmals kleinerer Adressraum (im lokalen Bereich), kein Tracking eines Knotens möglich
 - Nachteil: System zur dyn. Adressvergabe erforderlich, Grenzen der „Nachbarschaften“ müssen definiert werden

Adressierungssysteme

- Adresse für jeden Knoten unabdingbar für Routing
- Adressvergabesysteme können nach Eindeutigkeit der Adressen klassifiziert werden
 - **Global eindeutig**
 - **Netzwerkweit eindeutig**
 - **Lokal eindeutig**

Routing-Protokolle - Klassifizierung

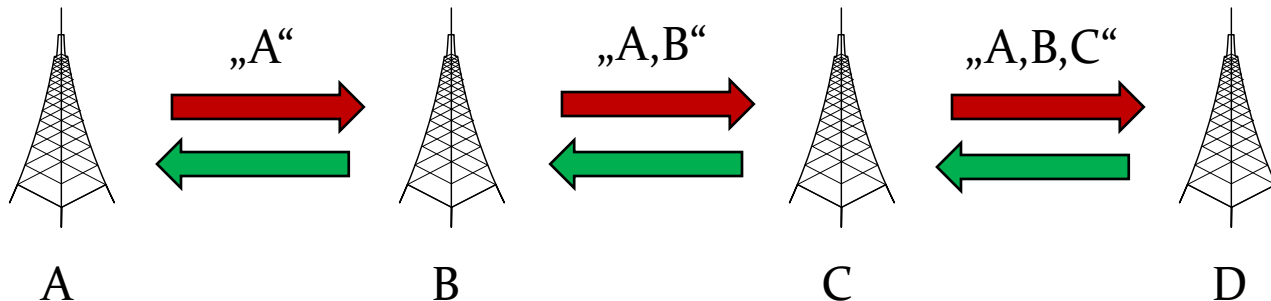
- Es gibt über 70 Ansätze für Routing-Protokolle
 - Viele dieser Protokolle wurden niemals implementiert
- Die Protokolle können in zwei Kategorien unterteilt werden
 - Reaktive Routingverfahren
 - Ermitteln Topologieinformationen erst dann, wenn diese für das Routing eines Datenpakets benötigt werden
 - Proaktive Routingverfahren
 - Ermitteln die komplette Netzwerktopologie bereits vor dem Versand von Datenpaketen

Reaktives Verfahren: DSR

- DSR (Dynamic Source Routing) platziert in jedem Paket die komplette, vorberechnete Route, die das Paket nehmen soll
- Die Route erhält der Sender aus einem Route Cache
- Ist noch keine Route zum gewünschten Ziel im Cache gespeichert, dann startet die „Route Discovery“-Phase

DSR – Route Discovery

- A möchte Paket an D senden
- A sendet einen „Route Request“-Broadcast
- B empfängt den Request, fügt seine Adresse hinzu und sendet ihn per Broadcast weiter
- C empfängt den Request; handelt wie B
- D empfängt Request von C, stellt fest, dass er das Ziel des Requests ist, erstellt „Route Reply“ und sendet diese zurück
- Rückweg wird entweder ebenfalls per „Route Request“ bestimmt oder die erhaltene Reihenfolge wird umgedreht

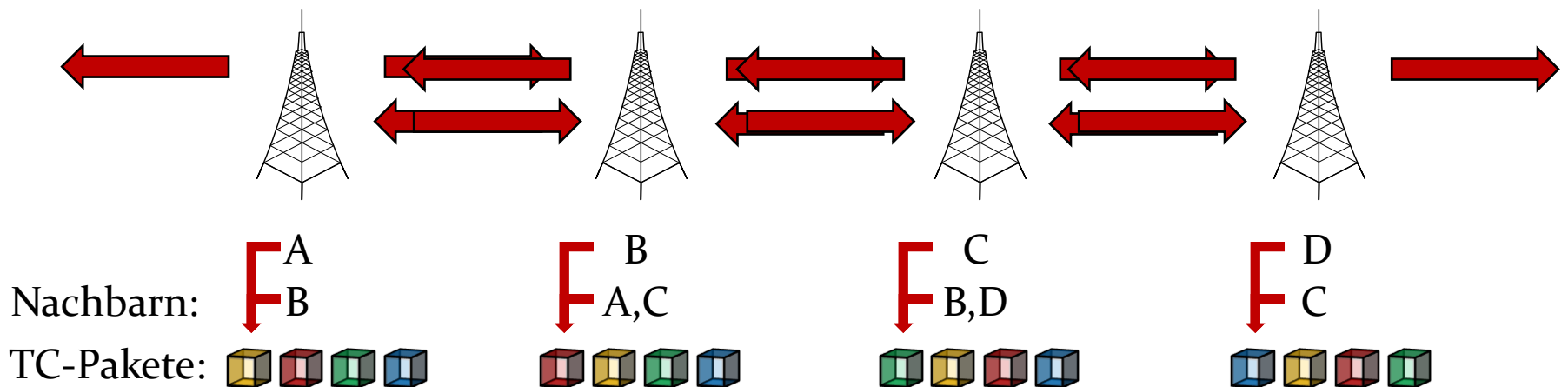


Proaktives Verfahren: OLSR

- OLSR (Optimized Link State Routing) benötigt keine Routing-Information im Datenpaket selbst
- Jeder Knoten kennt die komplette Netzwerktopologie
- Die Erkundung der Topologie basiert auf dem Link-State-Algorithmus
- OLSR fügt Optimierungen hinzu

OLSR – Klassisches Link-State-Verfahren

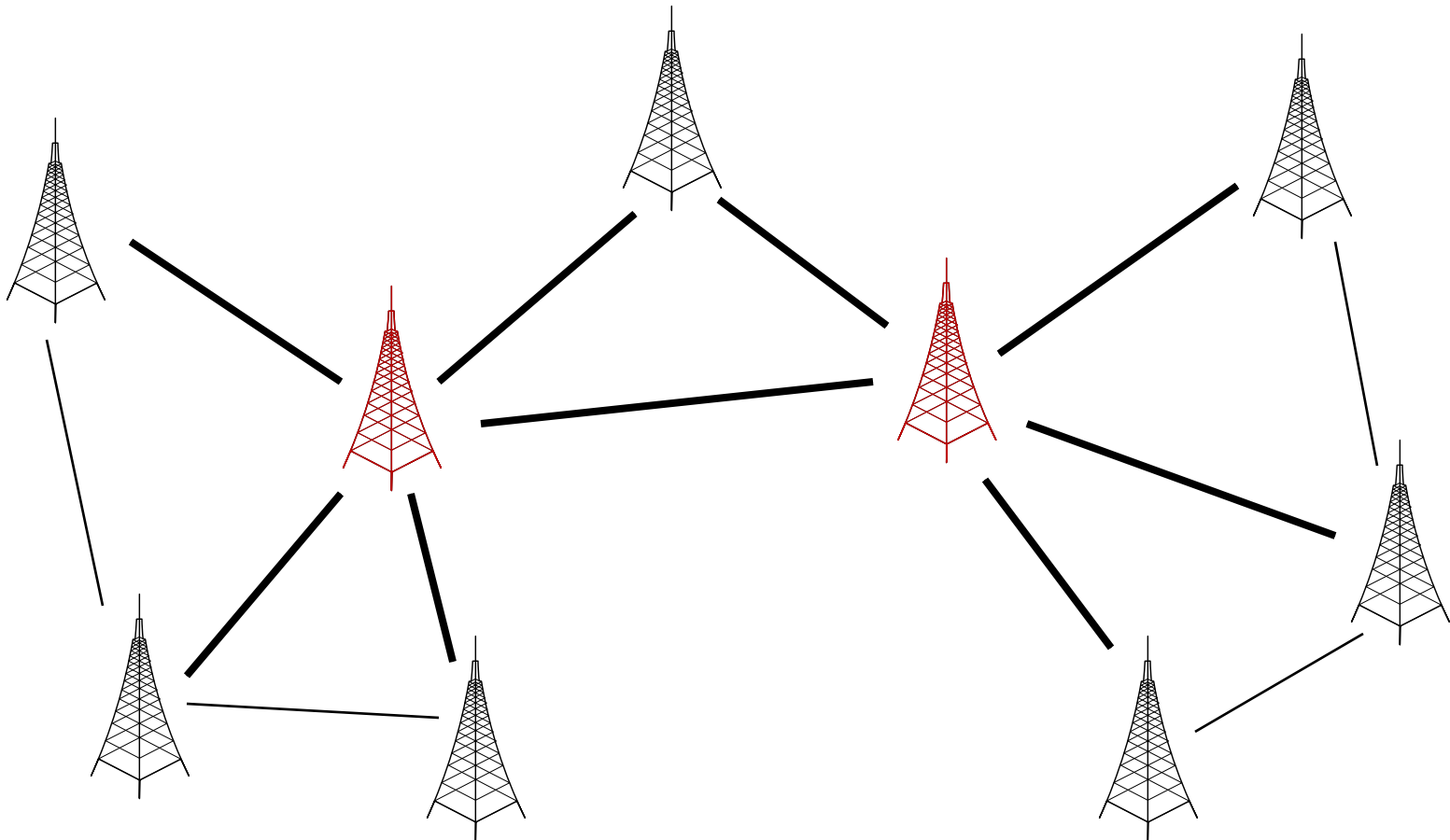
- Jeder Knoten sendet „Hello“-Nachrichten per Broadcast, erhält Antworten der direkten Nachbarn
- Jeder Knoten erstellt ein „Topology Control“-Paket mit seiner eigenen Adresse und seinen Nachbarn
- TC-Pakete werden per Broadcast und netzweiter Weiterleitung (Flooding) an alle Knoten verteilt
- Aus TC-Paketen kann ein Topologiebaum erstellt werden



OLSR - Optimierungen

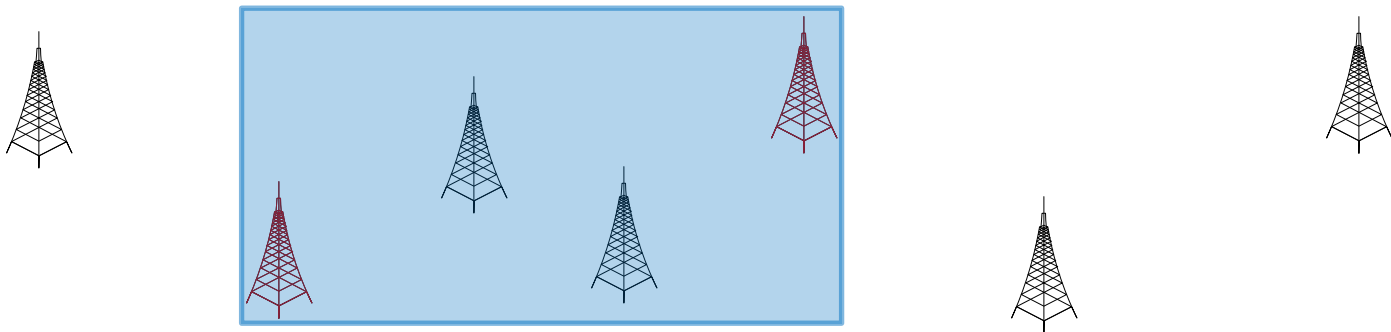
- Großes Problem des klassischen Algorithmus: durch Flooding über viele Knoten hohe Netzbelastung und schlechtes Skalierungsverhalten
- OLSR fügt eine weitere Hierarchieebene ein: die Multipoint-Relays (MPRs)
- Jeder Knoten wählt ein MPR-Set aus seinen direkten Nachbarn so, dass er darüber alle Knoten erreichen kann, die 2 Hops entfernt sind
- Die MPRs bilden den organisatorischen „Backbone“ des Netzes, sie erstellen die TC-Nachrichten und verteilen sie

OLSR – Multipoint Relays



Weitere Optimierungsmöglichkeiten

- Die große Zahl existenter Protokolle und Protokollvorschläge enthält weitere Ideen zur Optimierung der grundlegenden Protokollansätze
- Beispiel: LAR (Location Aided Routing)
 - Basiert auf DSR
 - Broadcast-Nachrichten werden aber auf eine geografische Region beschränkt



Datenintegrität

- Die Integrität (gesendete Daten = empfangene Daten) eines Datenpakets ist durch die Weiterleitung über Knoten eines i.d.R. unbekanntem Besizers gefährdet
- Integritätsschutz kann durch Ende-zu-Ende-Verschlüsselung stattfinden (dazu später mehr)
- Aber: wenn auch routende Teilnehmer eine Nachricht lesen können sollen (Beispiel: Car2Car) ist Verschlüsselung ungeeignet

 **Digitale Signatur**

Datenintegrität – Digitale Signatur

- Durch Signatur jedes Pakets beim Sender kann die Integrität sowohl von den weiterleitenden als auch dem empfangenen Knoten geprüft werden
- Voraussetzung: Public Key des Senders bekannt
- Lösungsmöglichkeiten:
 - Jedem Netzknoten eigenes Public/Private-Key-Paar geben. Public-Key-Austausch sicher durchführen.
 - Austausch über alternatives Medium (z.B. per Hand, GSM)
 - Feste Vergabe eines Public Key für einen Key-Server, weitere Keys aller Teilnehmer vom Key-Server erhältlich
 - Feste Keys in Endgeräte einbauen (z.B. Car2Car)

Datensicherheit

- Sicherheit auf dem Transportweg = Verschlüsselung
- Typischerweise gewünscht: Ende-zu-Ende-Sicherheit
- Problem dabei: Authentifikation
 - Schlüsselaustauschprotokolle wie Diffie-Hellman funktionieren auch in Mesh-Netzen
 - Die Authentizität des Kommunikationspartners muss aber zweifelsfrei sichergestellt werden können
- Lösungsansätze ähnlich „Datenintegrität“
 - Authentifikation beim Schlüsselaustausch ist ein Integritätsproblem

Privatsphäre

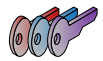
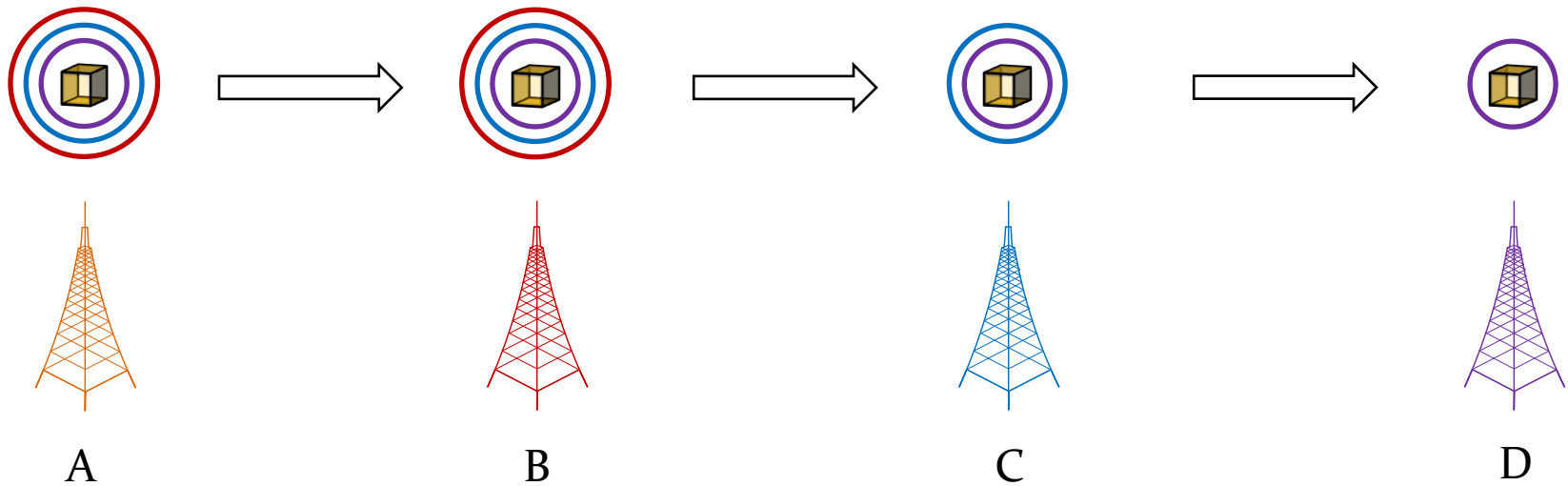
- Selbst bei Ende-zu-Ende-Verschlüsselung können die Verkehrsinformationen mitgelesen werden
- Um dies zu vermeiden, muss Anonymität hergestellt werden

 **Onion Routing**

Onion Routing – Einführung

- Onion Routing erlaubt eine vollständige Anonymisierung der Übertragung
- Die Zuordnung zwischen Sender und Empfänger ist nur dann herstellbar, wenn alle Zwischenknoten zusammenarbeiten
- Erreicht wird dies durch mehrfache Verschlüsselung

Onion Routing



Public Keys von B,C,D

- kennt A
- kennt C

- kennt B
- kennt D

Weder B noch C kennt Sender und Empfänger!

Onion Routing

- Anonymität bei der Datenübermittlung sowie Datenintegrität und -sicherheit können durch Onion Routing gewährleistet werden
- Statt zuvor verteilter Public Keys können symmetrische Keys vom Sender bestimmt und zunächst an die Knoten verteilt werden
 - Senkt CPU-Belastung durch asymm. Kryptographie
 - Erfordert aber nach wie vor eine Verteilung des Public Key jedes Nodes

Onion Routing – die perfekte Lösung?

- Onion Routing bietet
 - Anonymität
 - Datensicherheit
 - Datenintegrität
- Onion Routing erfordert aber eine Mindestanzahl von 2 Hops zwischen Sender und Empfänger
- Außerdem erzeugt Onion Routing Overhead...
 - ...bei der Datenübertragung selbst
 - ...beim Schlüsselaustausch/Aufbau der Verbindung
- Public/Private-Key-Infrastruktur erforderlich

Noch zu lösende Problemfelder

- Fehlende Standardisierung
 - Es gibt zwar RFCs zu mehreren Protokollen, aber diese sind „experimental“
 - Die Vielzahl an vorgeschlagenen, oft wenig getesteten und zueinander inkompatiblen Protokollen beeinträchtigt die Inoperabilität
- Die Stabilität selbst der bereits eingesetzten Protokolle in sehr großen Netzwerken ist noch wenig bis gar nicht praktisch getestet
- Die Energie-Problematik bei akkubetriebenen Mobilgeräten ist noch weitgehend ungelöst
- Häufig stellt die Public/Private-Key-Infrastruktur ein Problem dar

Quellen

- Holger Karl, Andreas Willig, **Protocols and Architectures for Wireless Sensor Networks**, Wiley, 2007.
- David B. Johnson, David A. Maltz, Josh Broch, **DSR: The Dynamic Source Routing Protocol für Multi-Hop Wireless Ad Hoc Networks**, Computer Science Department, Carnegie Mellon University, Pittsburgh, 2001.
<http://www.monarch.cs.rice.edu/monarch-papers/dsrchapter00.pdf>
- D. Johnson (Rice University), Y. Hu (UIUC), D. Maltz (Microsoft Research), **The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks for IPv4**, RFC 4728, Februar 2007
- P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, **Optimized Link State Routing Protocol for Ad Hoc Networks**, Hipercom Project, INRIA Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France
- David Goldschlag, Michael Reed, Paul Syverson, **Onion Routing for Anonymous and Private Internet Connections**, Communications of the ACM, vol. 42, num. 2, Februar 1999
<http://www.onion-router.net/Publications/CACM-1999.pdf>
- http://en.wikipedia.org/wiki/Onion_routing