ScatterNetz-Routing (SNR)

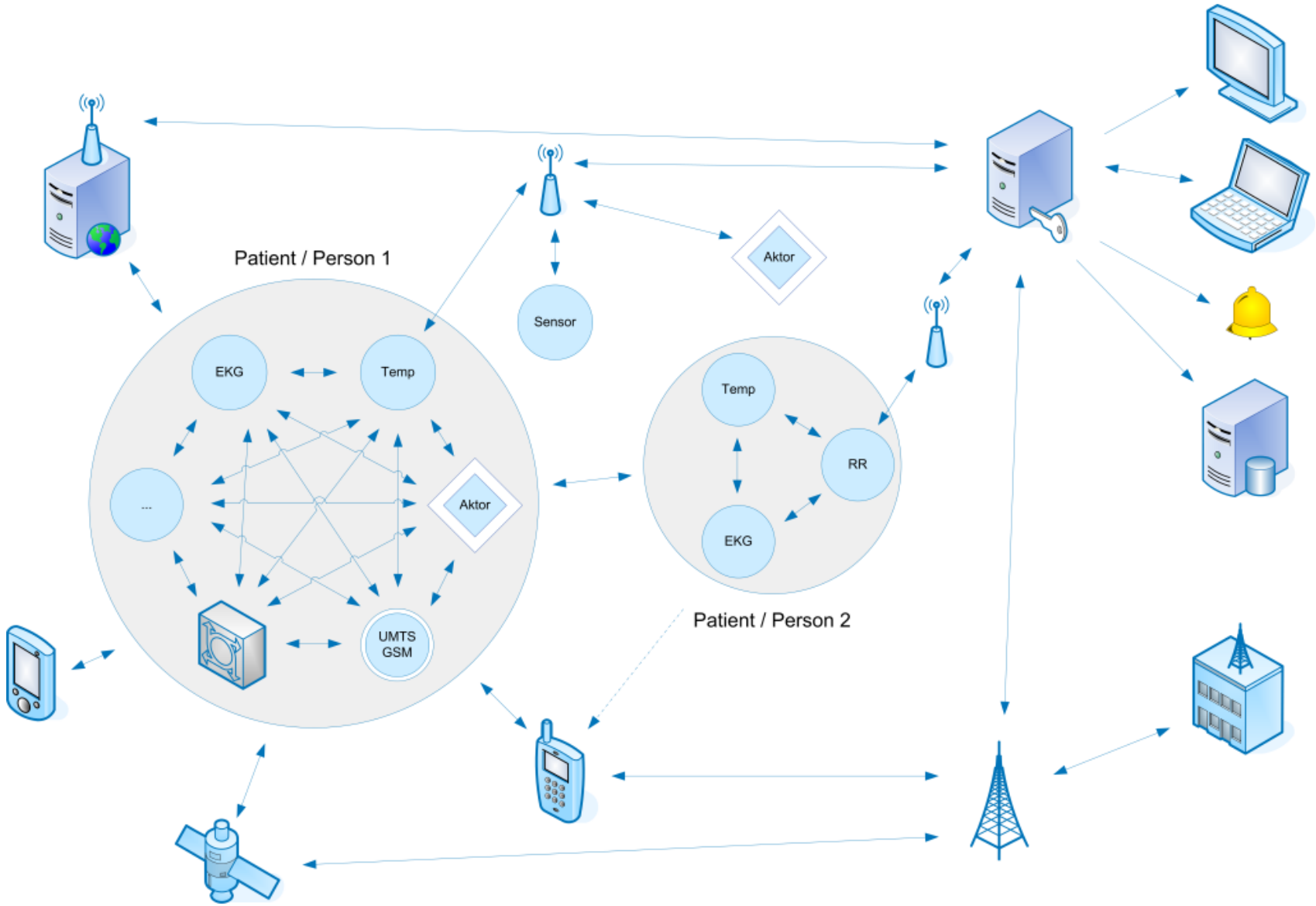
Multihopkommunikation für medizinische Bluetooth ad hoc Netzwerke

Andreas L. Kuntz

22. März 2006

Gliederung

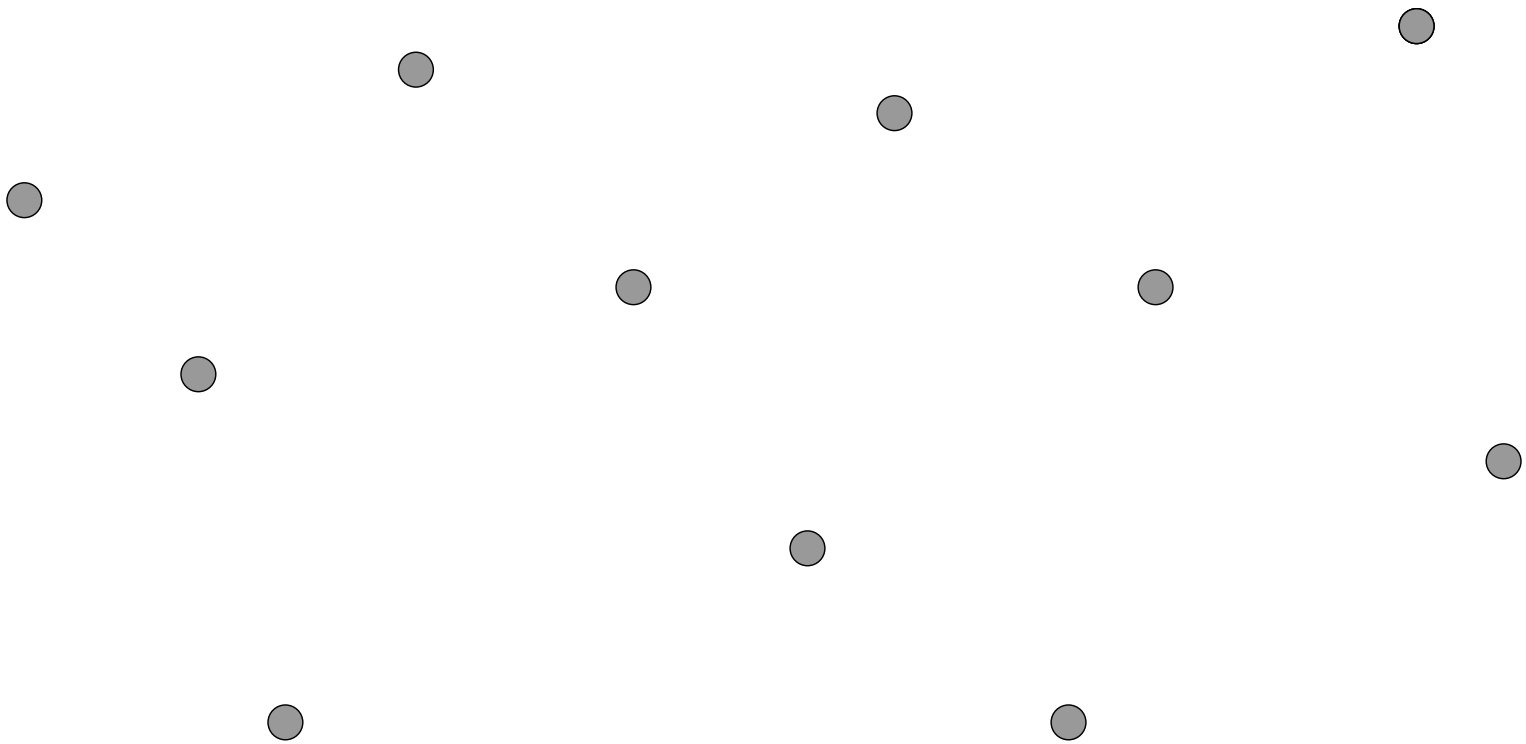
DaumeN


Patient / Person 1
Sensor
Aktor
EKG
Temp
...
Aktor
UMTS
GSM
Temp
RR
EKG
Patient / Person 2

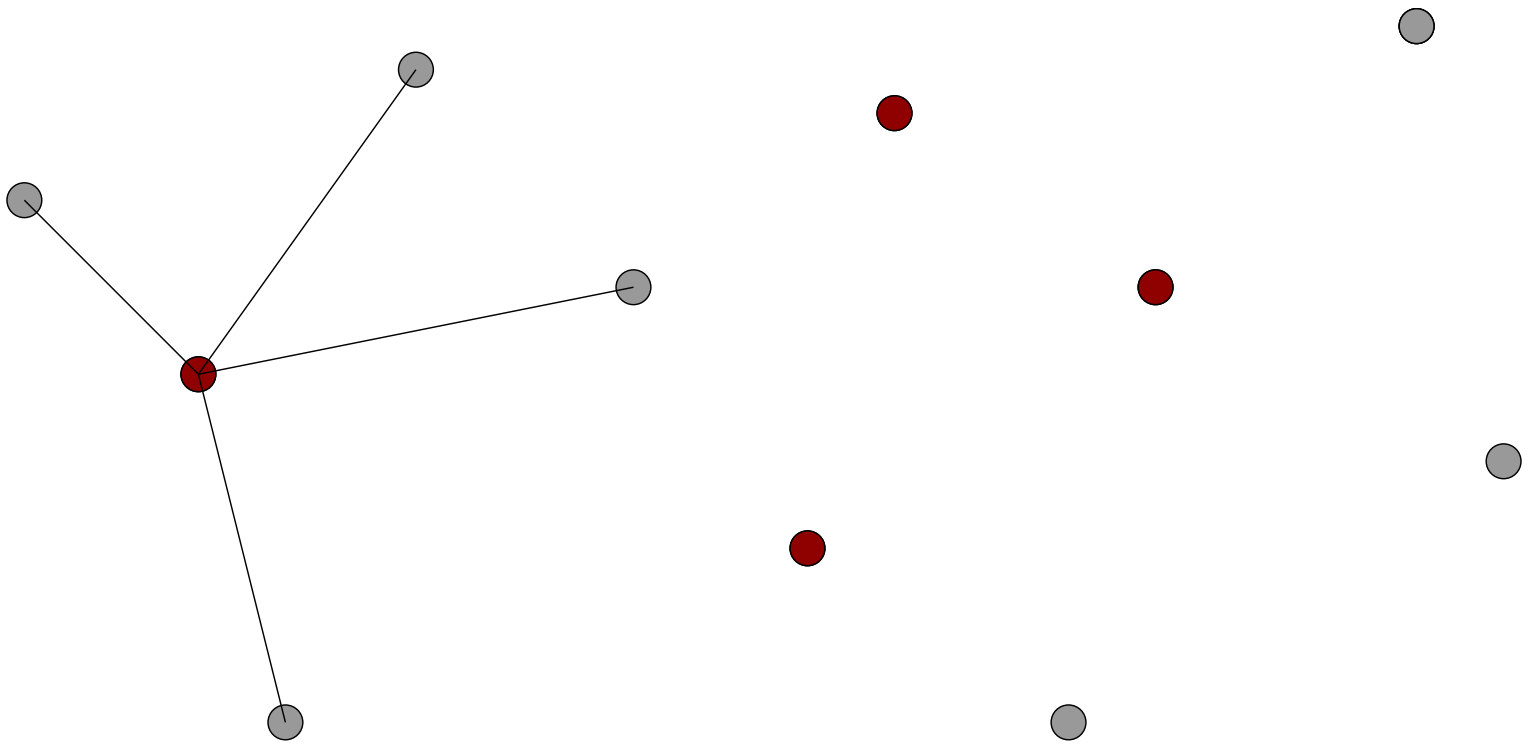
Anforderungen & Ziele

- Multihopkommunikation für Bluetooth
- Automatische Reparatur unterbrochener Pfade
- Ad hoc Fähigkeit
- Platformunabhängigkeit
- Funktionsfähigkeit in heterogenem Umfeld
- Keine Veränderungen gegenüber der Bluetooth Spezifikation

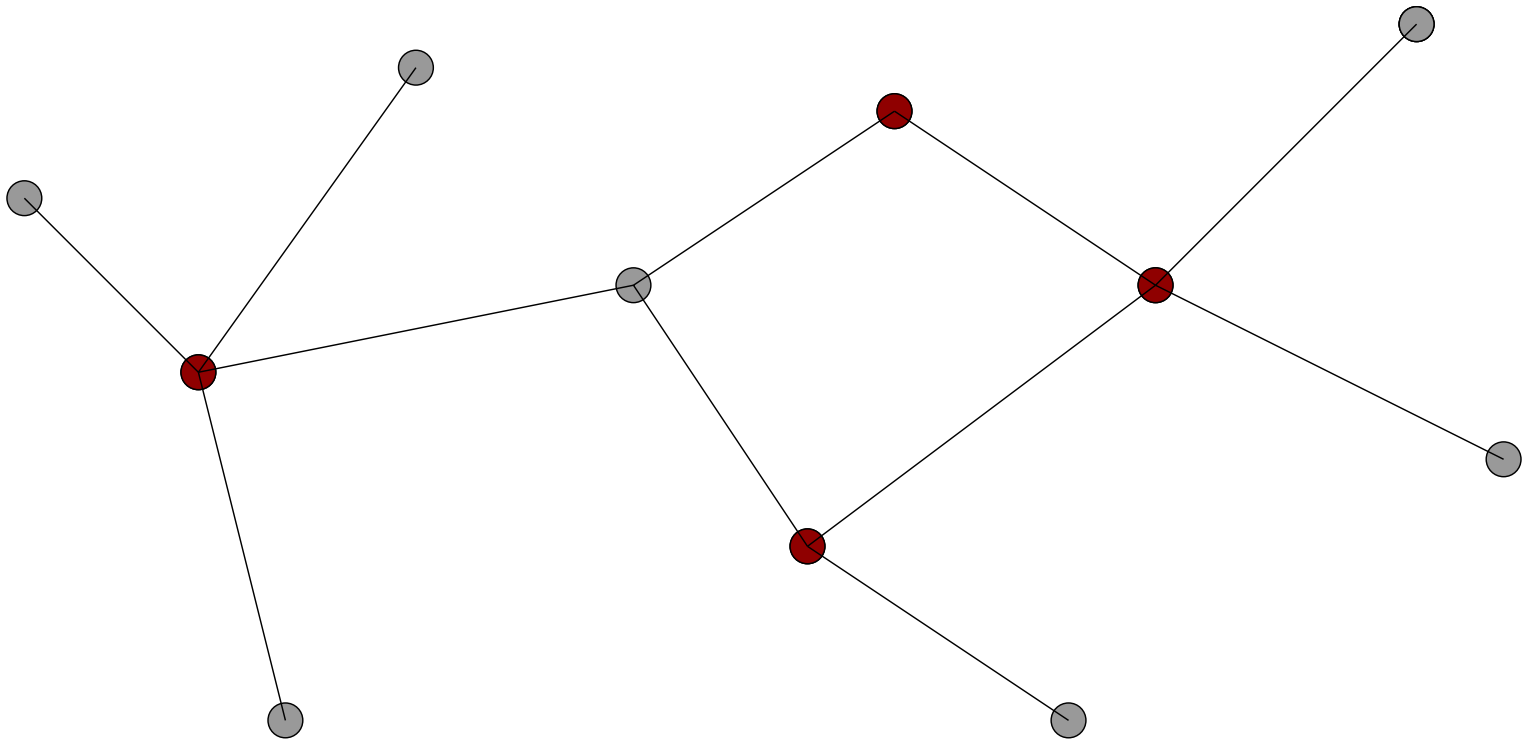
Schritte zur Multihopkommunikation SNR Überblick



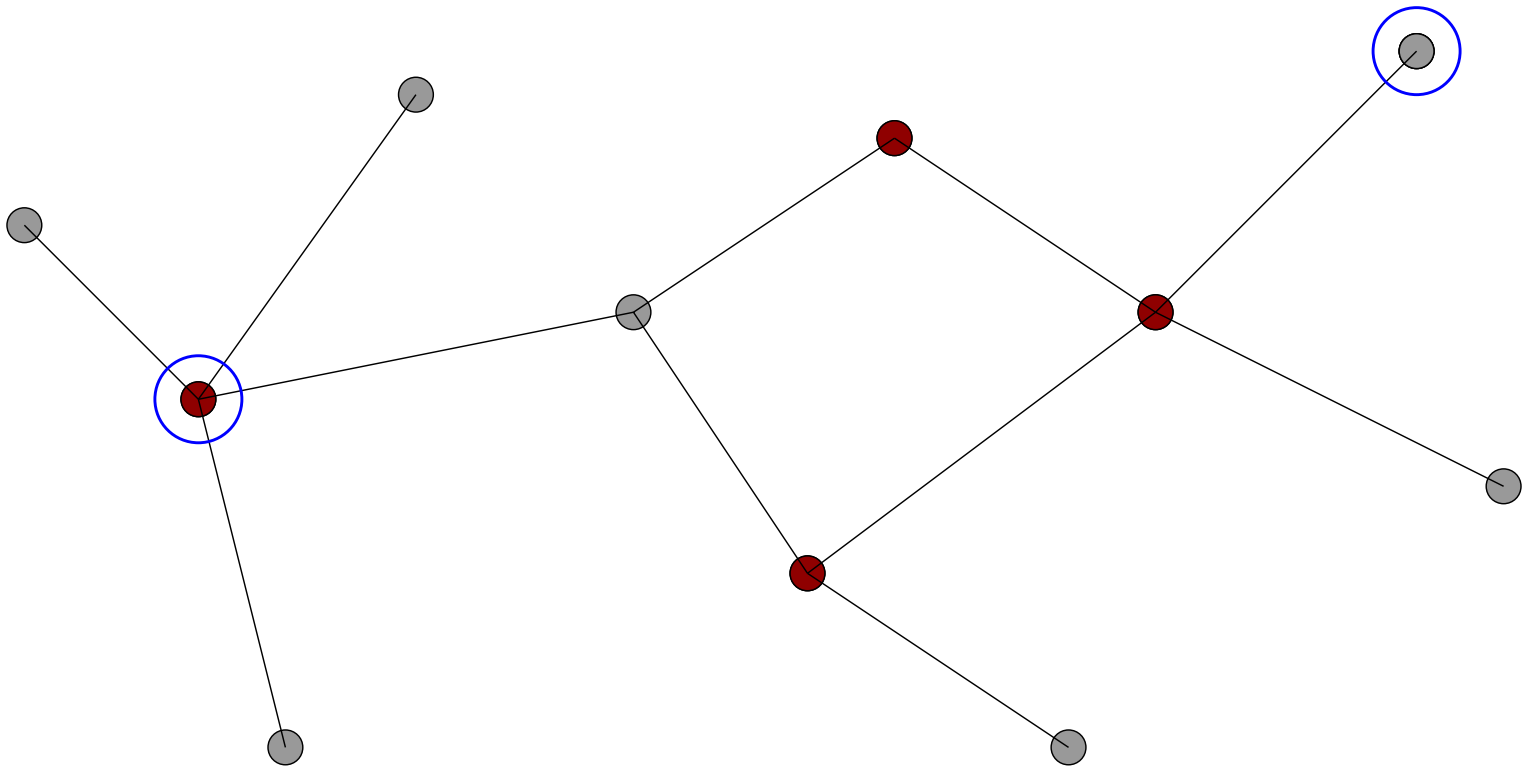
Schritte zur Multihopkommunikation SNR Überblick



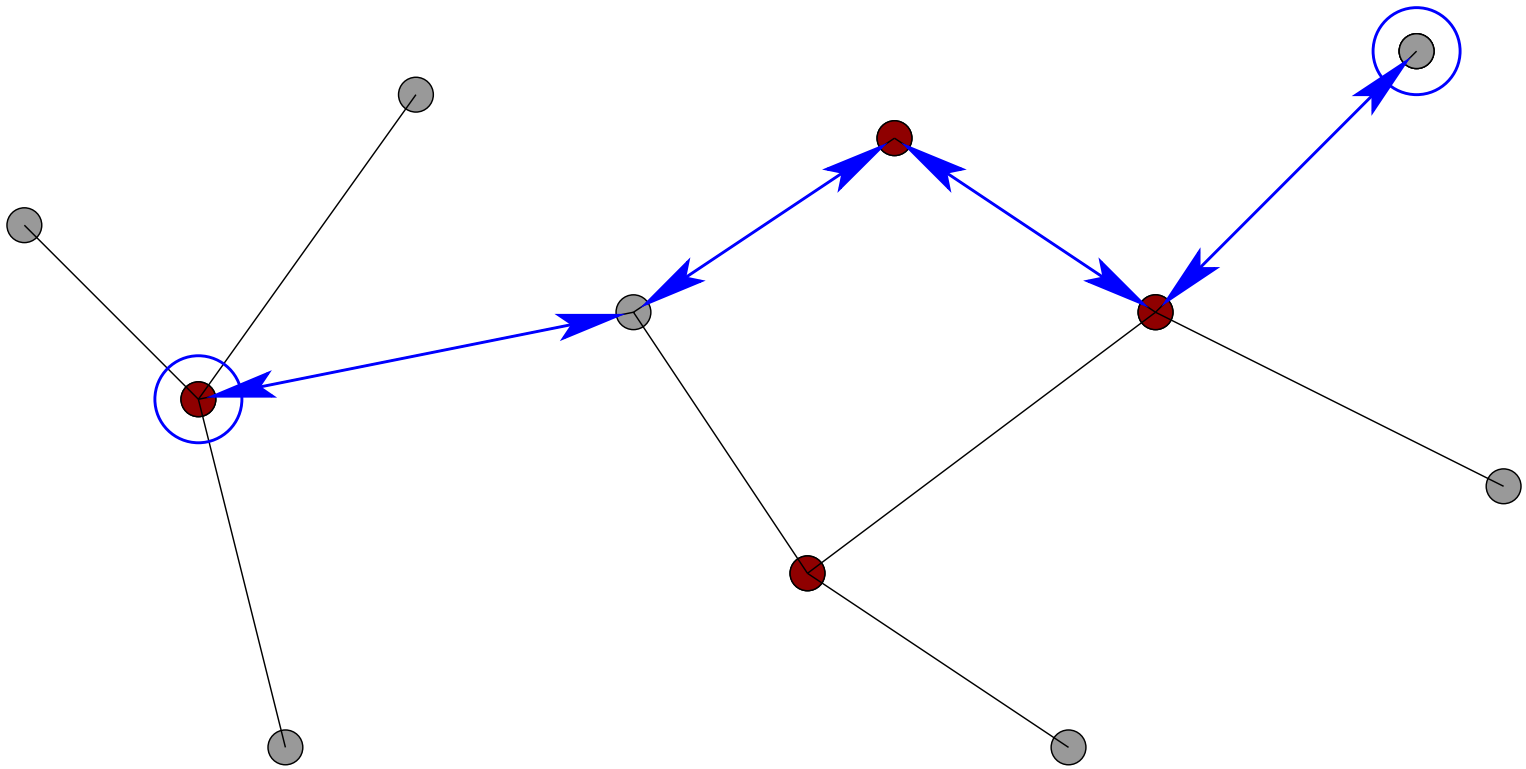
Schritte zur Multihopkommunikation SNR Überblick



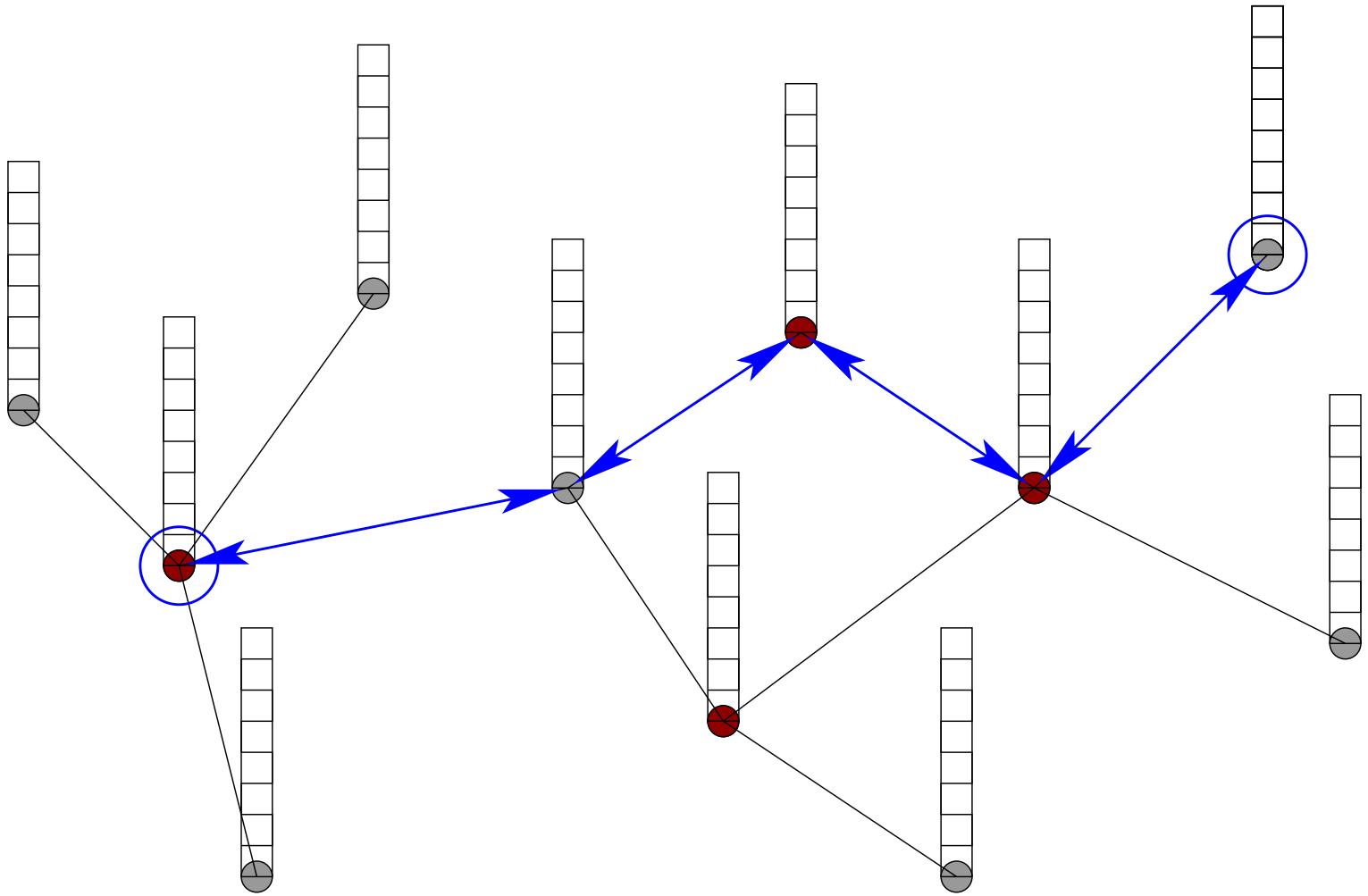
Schritte zur Multihopkommunikation SNR Überblick



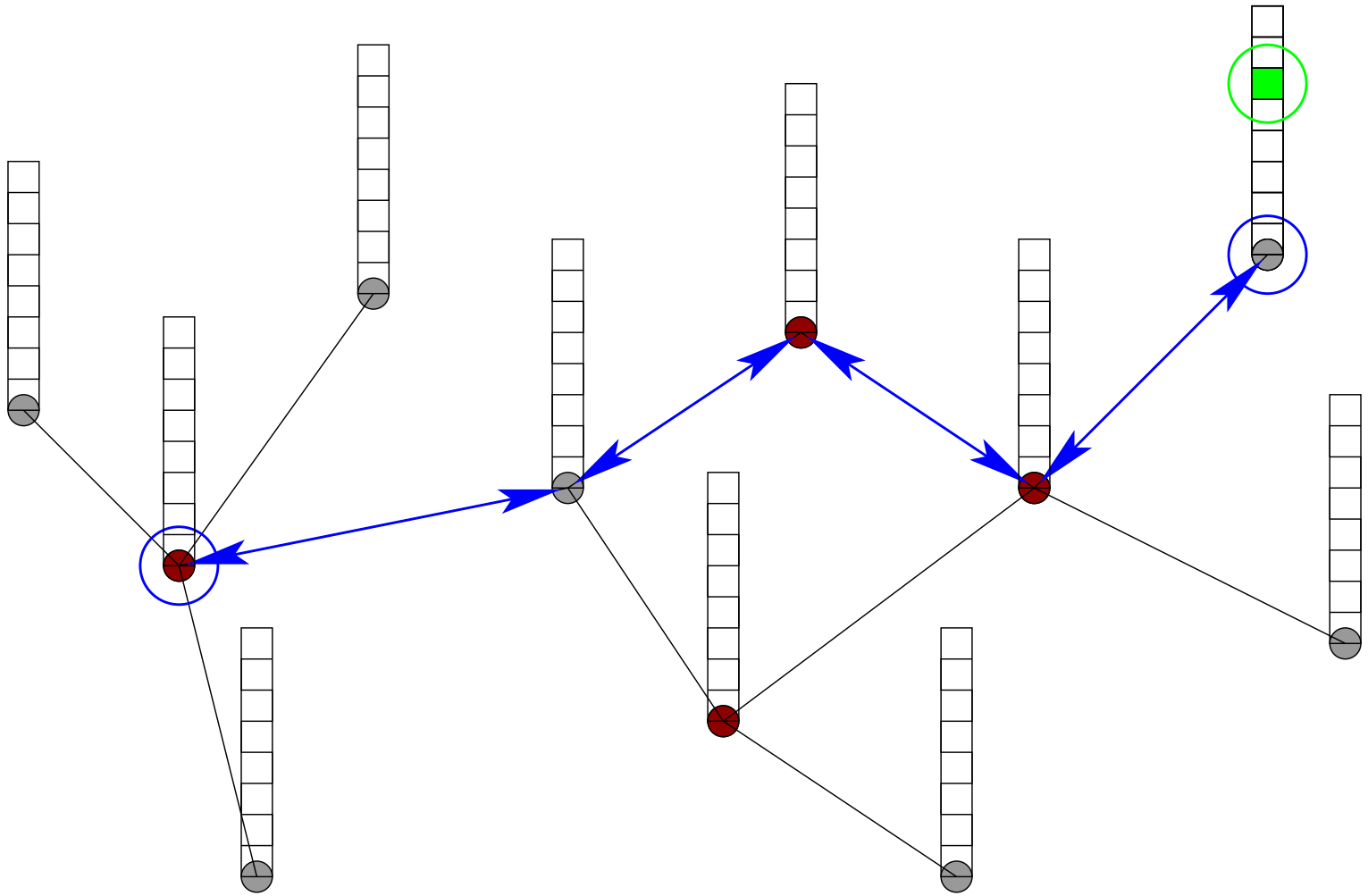
Schritte zur Multihopkommunikation SNR Überblick



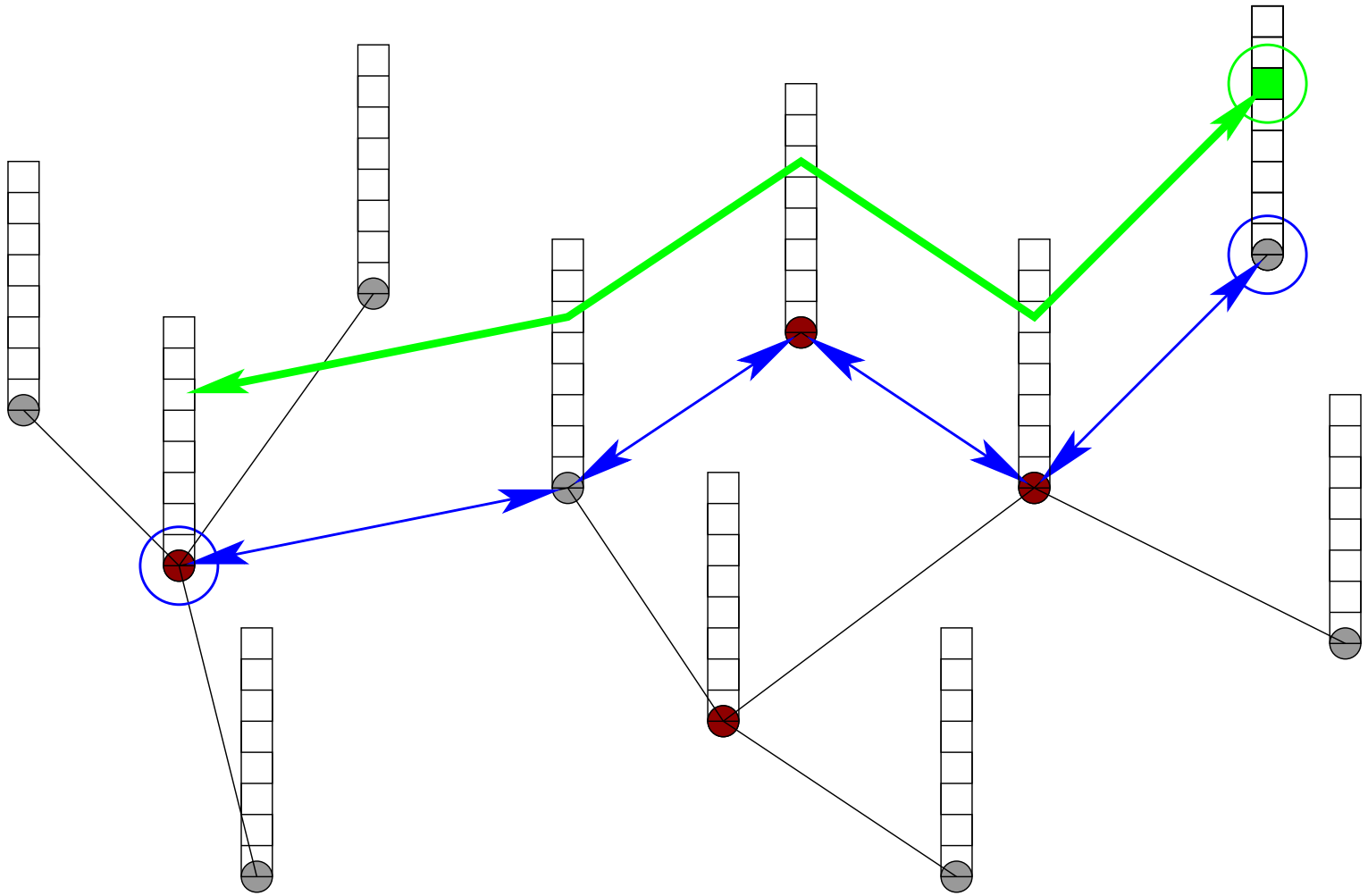
Schritte zur Multihopkommunikation SNR Überblick



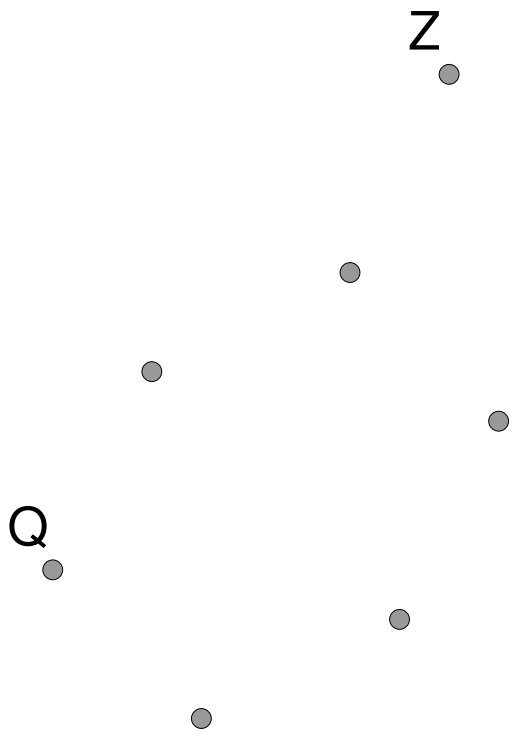
Schritte zur Multihopkommunikation SNR Überblick



Schritte zur Multihopkommunikation SNR Überblick



Wegfindung

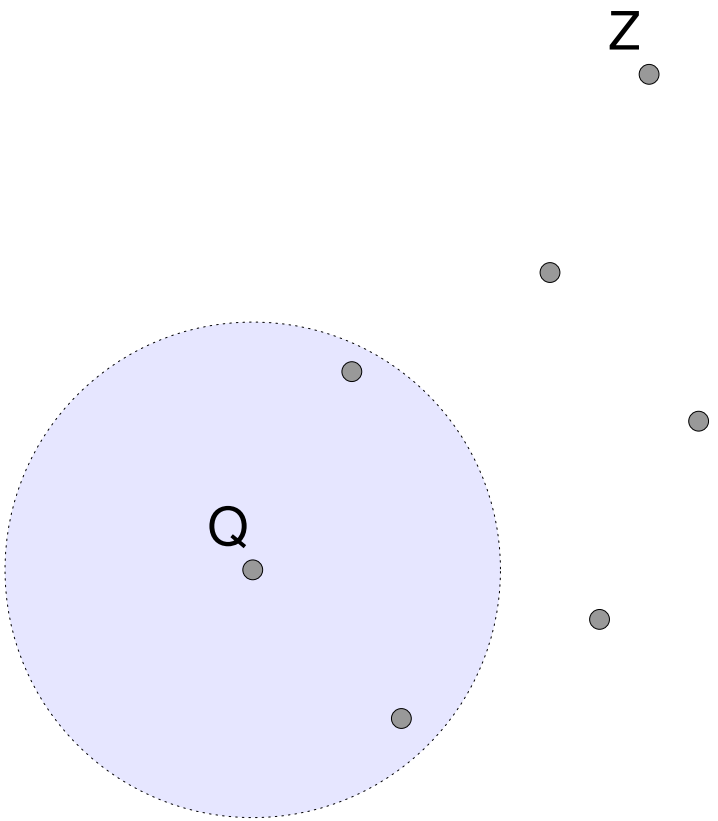


- Q sucht Pfad nach Z

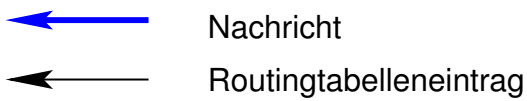
Nachricht

Routingtabelleneintrag

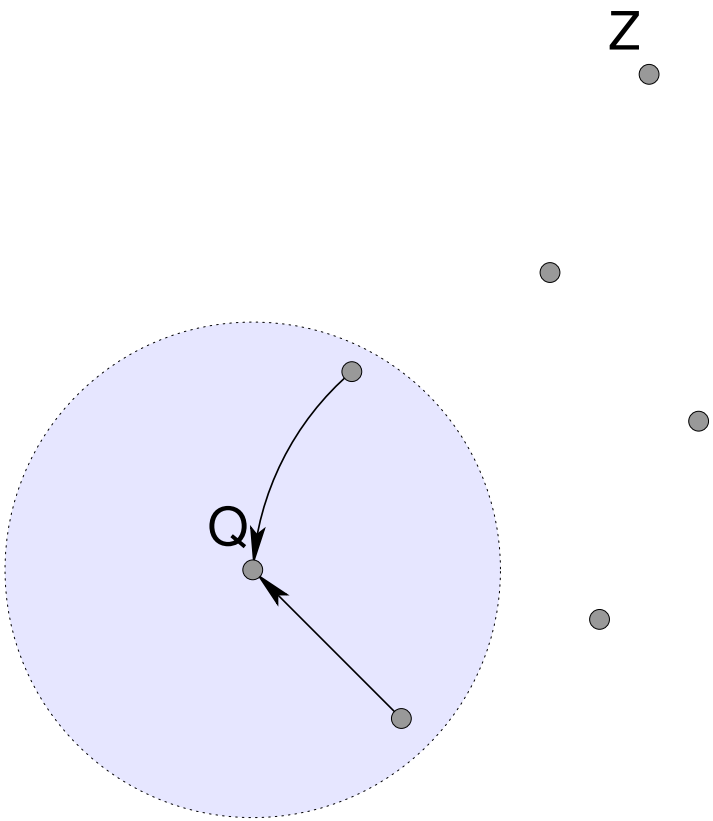
Wegfindung



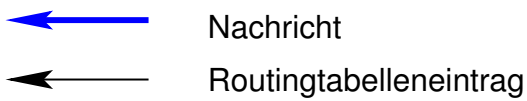
- Q sucht Pfad nach Z
- Anfrage fluten



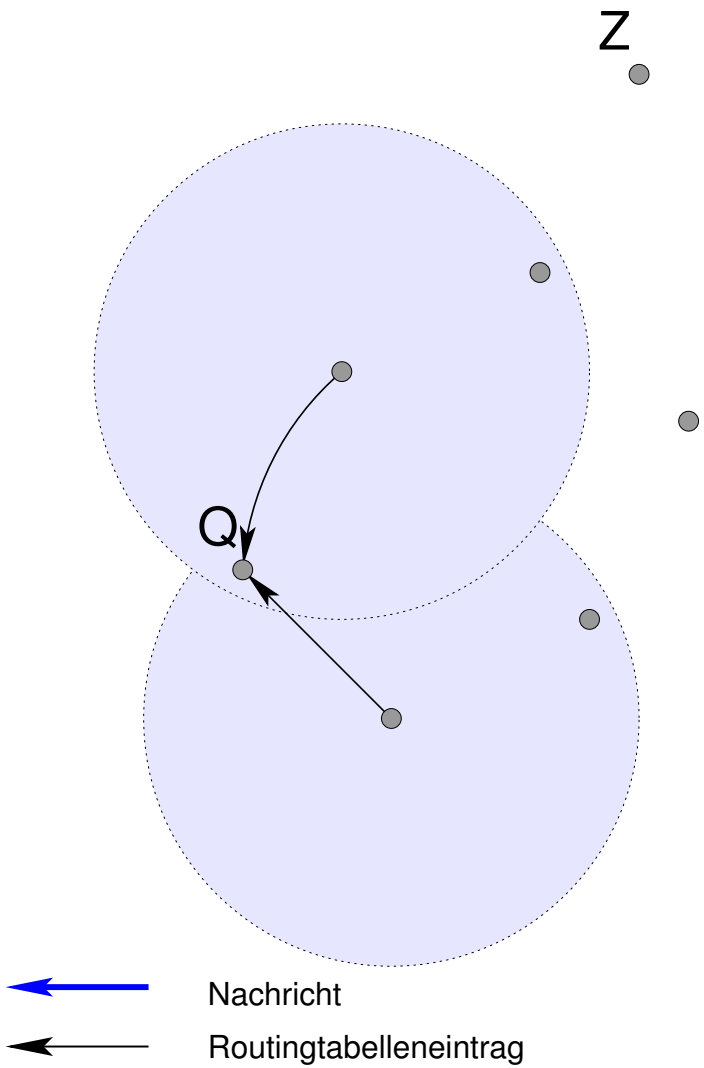
Wegfindung



- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein

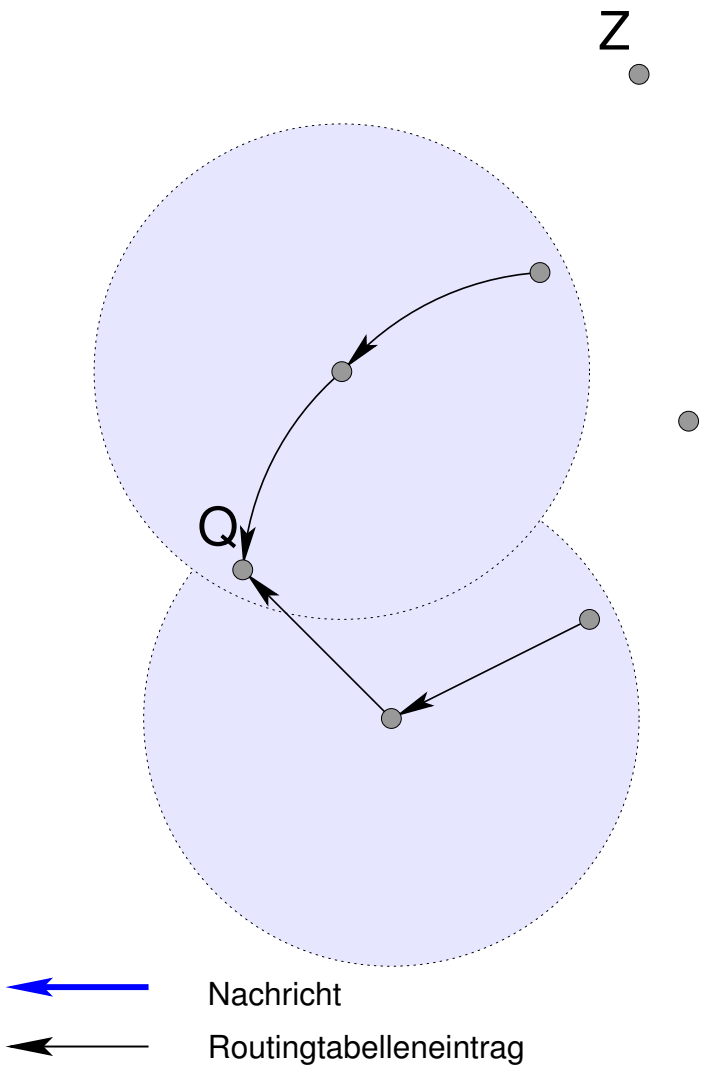


Wegfindung



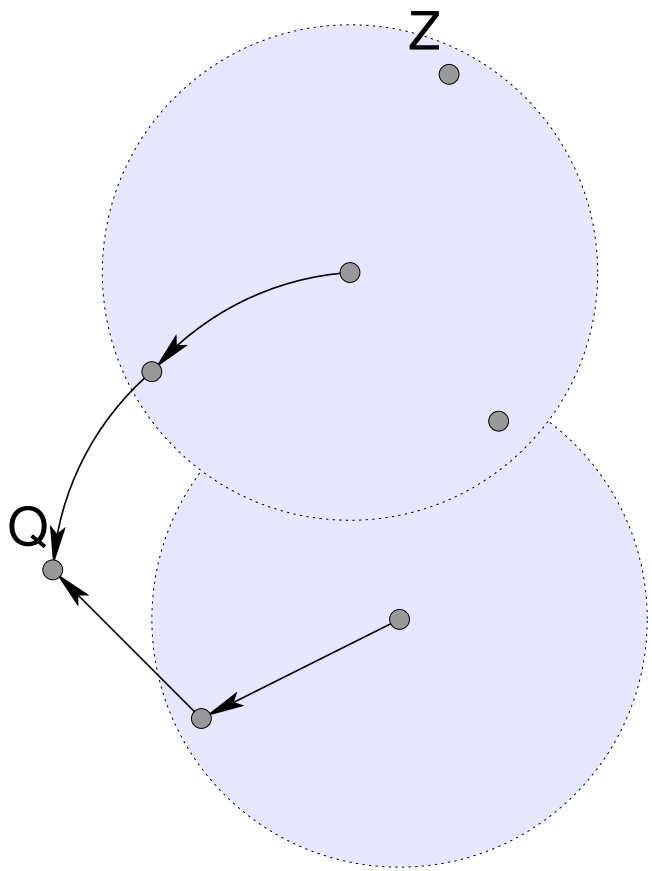
- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein

Wegfindung



- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein

Wegfindung

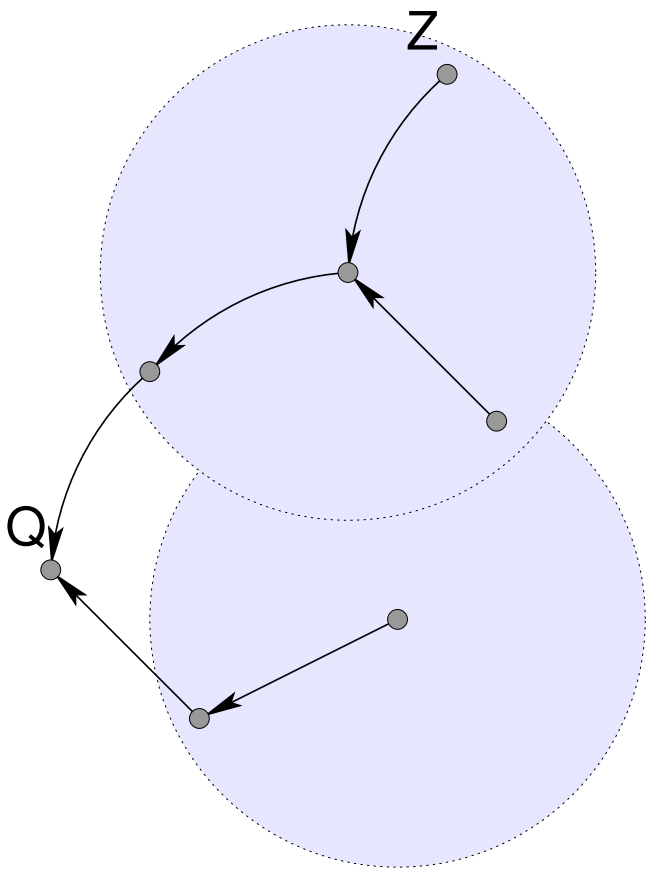


- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein

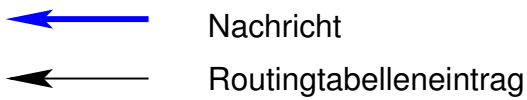
Nachricht

Routingtabelleneintrag

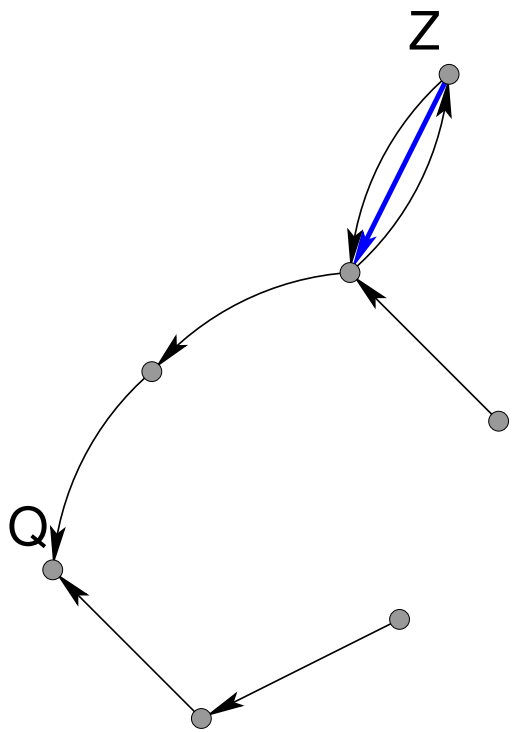
Wegfindung



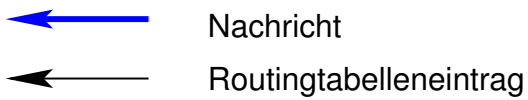
- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage



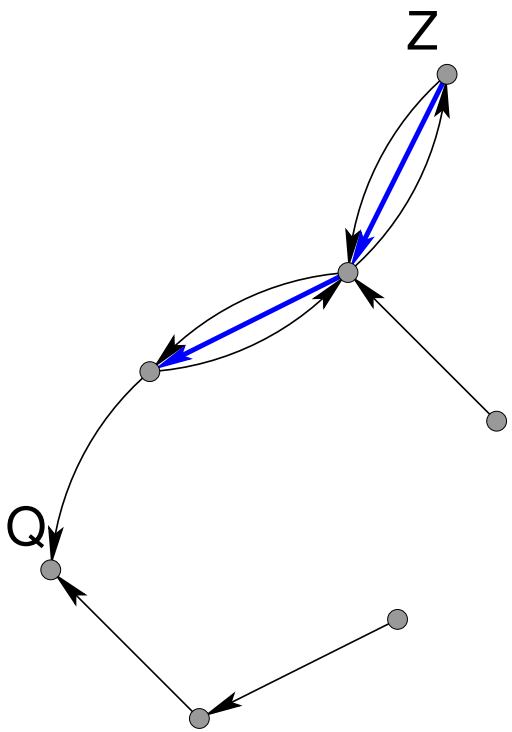
Wegfindung



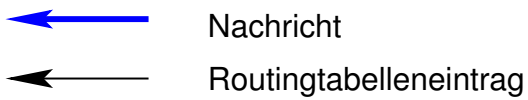
- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage
- Antwort senden
- Hinweg eintragen



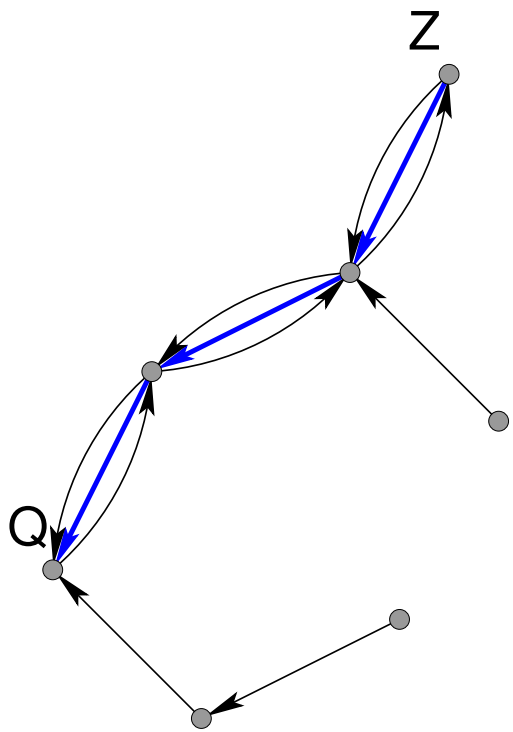
Wegfindung



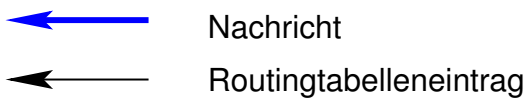
- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage
- Antwort senden
- Hinweg eintragen



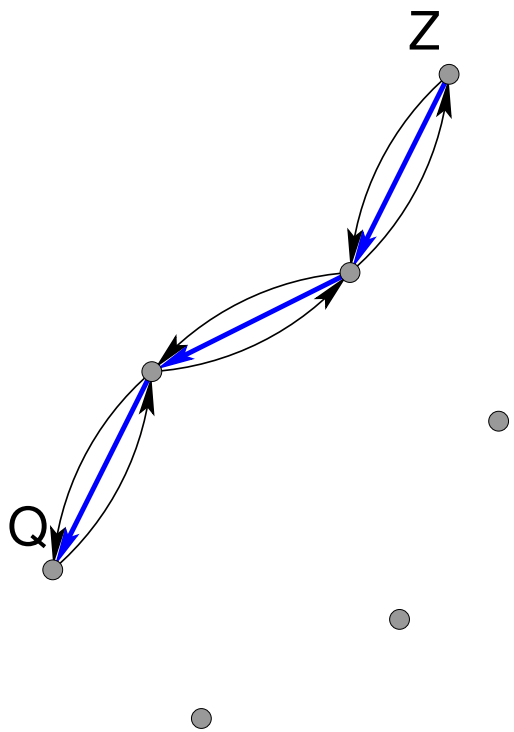
Wegfindung



- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage
- Antwort senden
- Hinweg eintragen
- Q empfängt Antwort
- Pfad gefunden!



Wegfindung

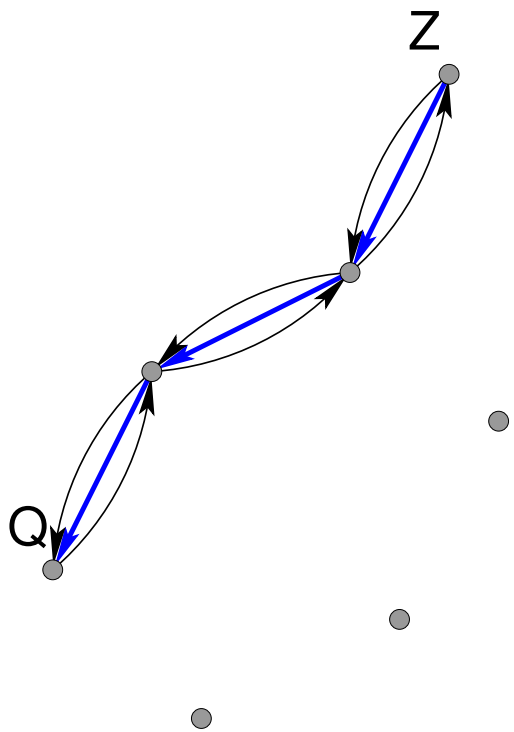


- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage
- Antwort senden
- Hinweg eintragen
- Q empfängt Antwort
- Pfad gefunden!
- Timeout für ungenutzte Verbindungen

Nachricht

Routingtabelleneintrag

Wegfindung



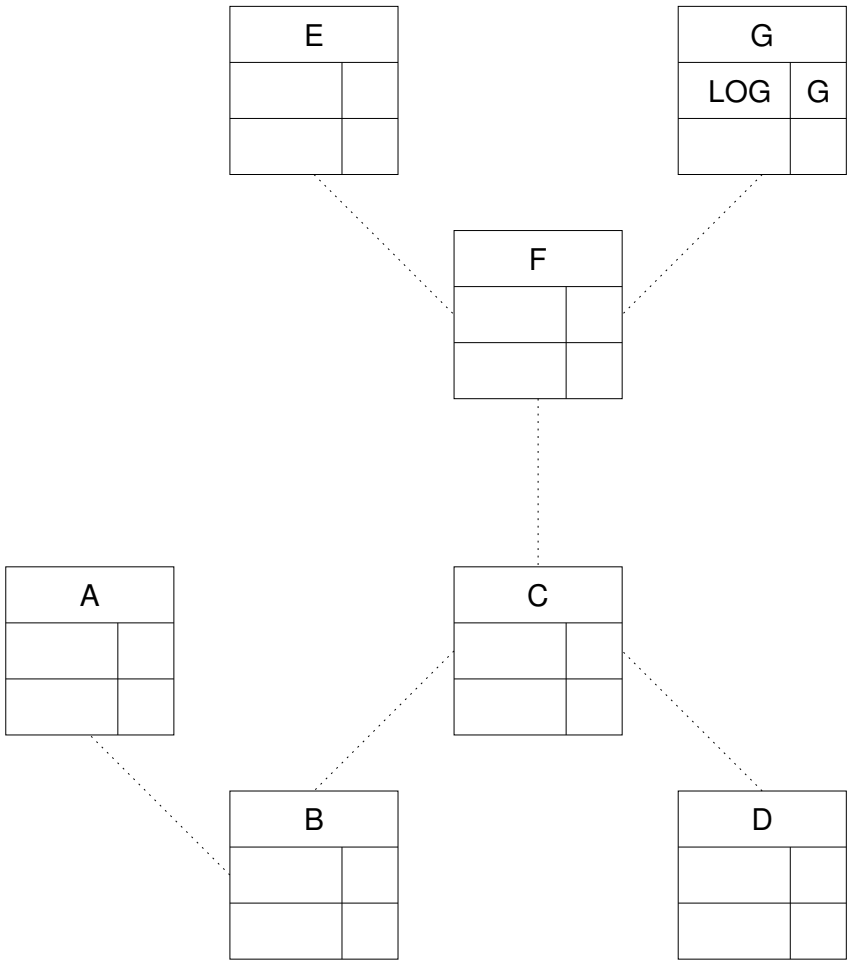
- Q sucht Pfad nach Z
- Anfrage fluten
- Empfänger trägt Rückweg ein
- Z empfängt Anfrage
- Antwort senden
- Hinweg eintragen
- Q empfängt Antwort
- Pfad gefunden!
- Timeout für ungenutzte Verbindungen

- Topologieausprägung
- Wegfindung

Nachricht

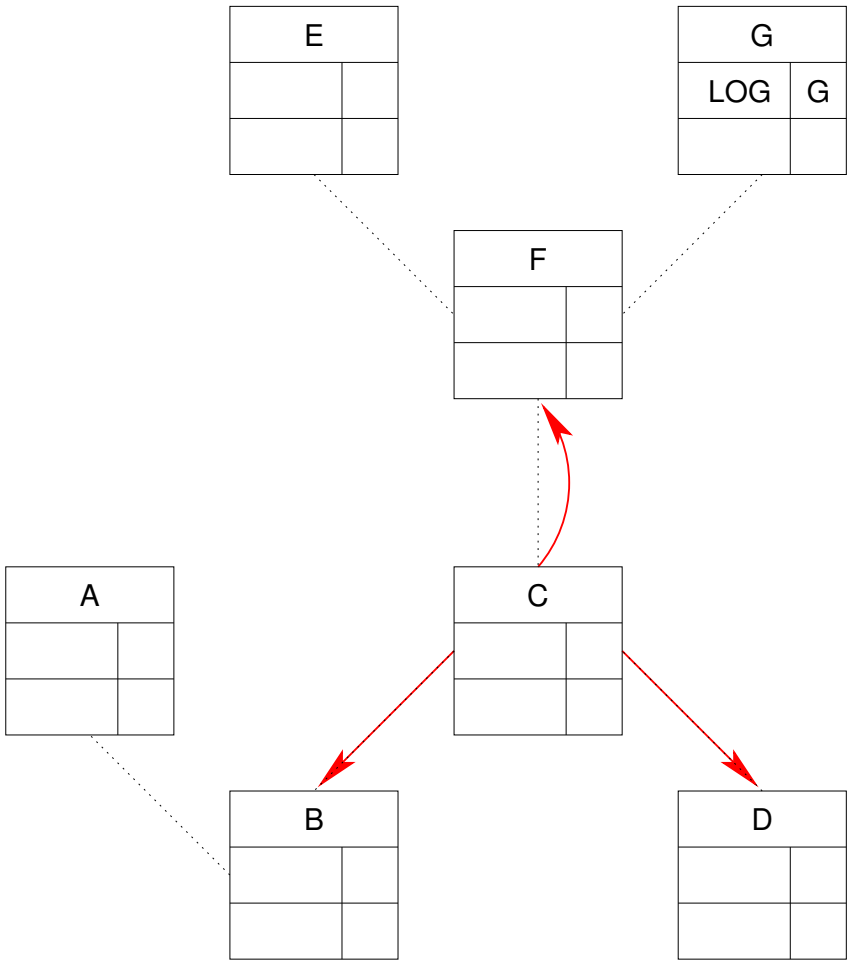
Routingtabelleneintrag

Dienstsuche



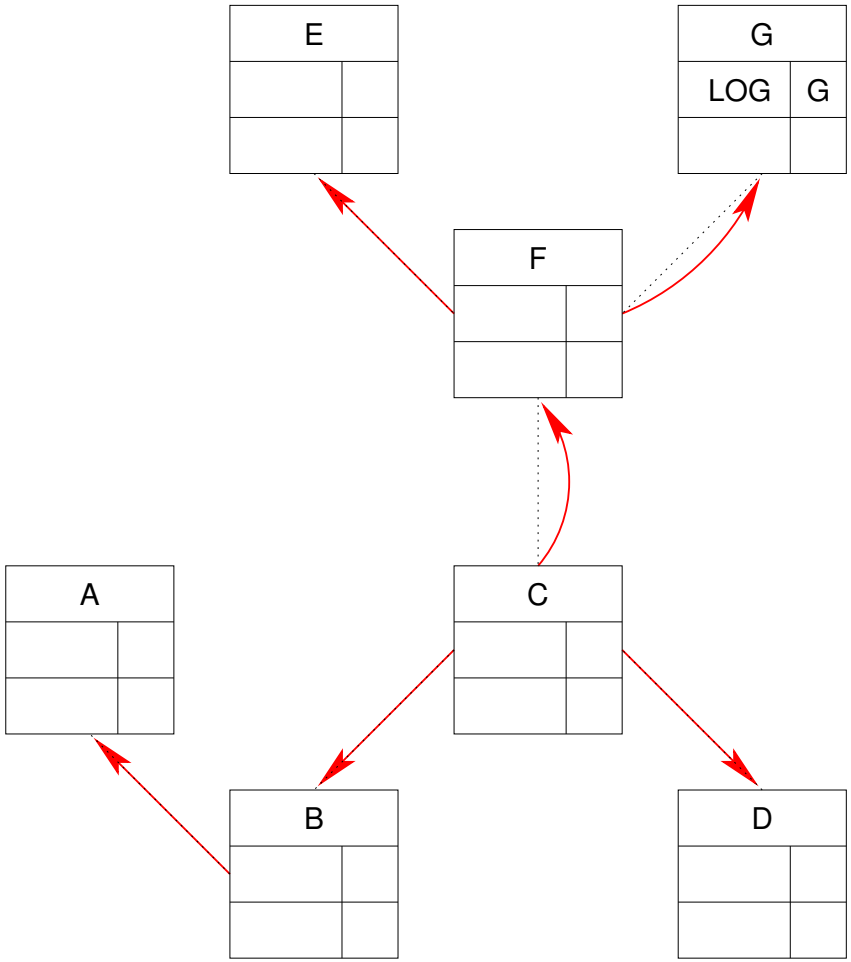
- C sucht Dienst LOG

Dienstsuche



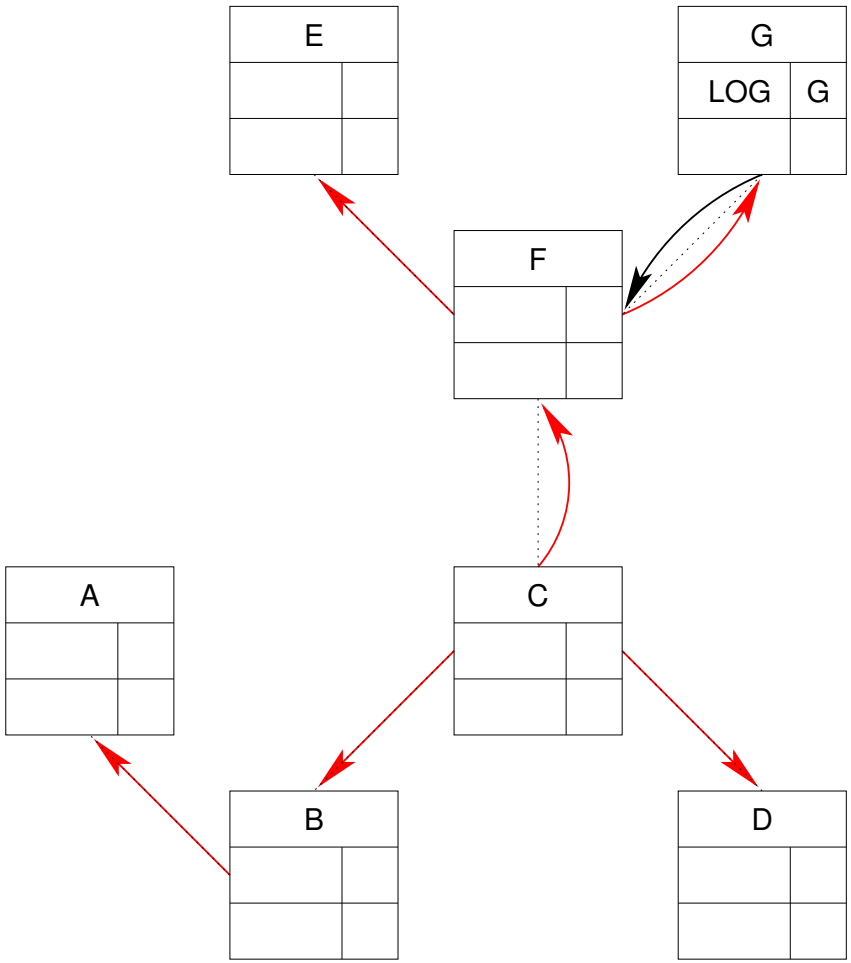
- C sucht Dienst LOG
- flute Anfrage REQ(LOG)

Dienstsuche



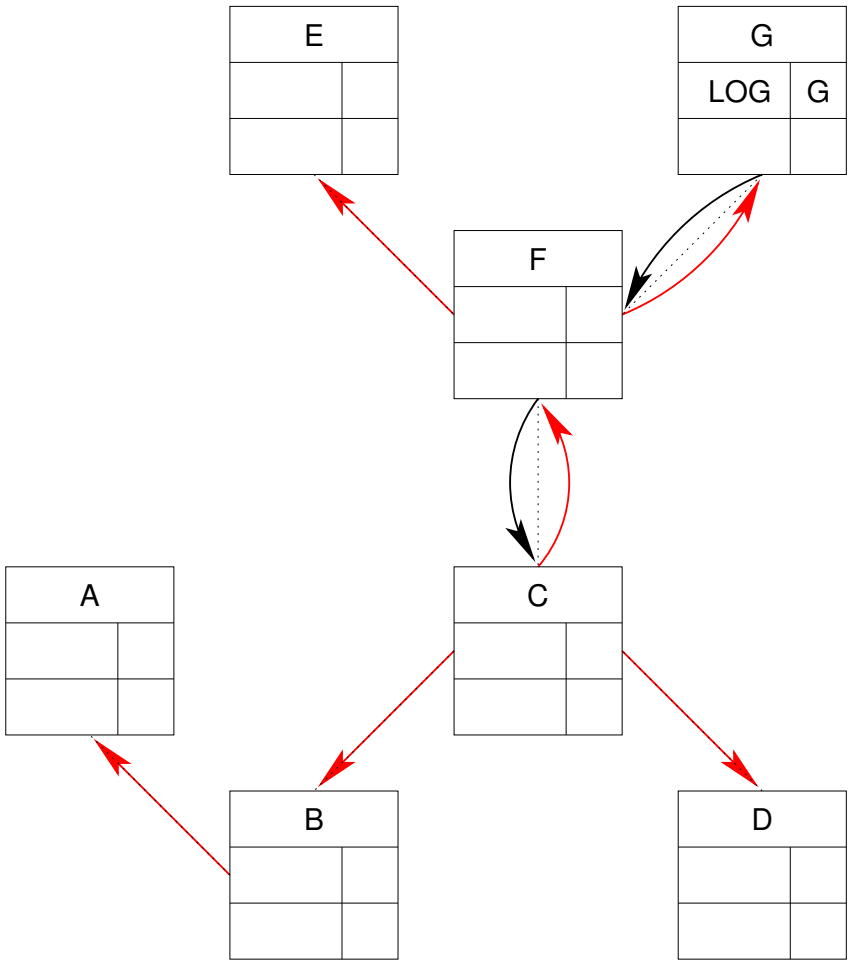
- C sucht Dienst LOG
- flute Anfrage REQ(LOG)

Dienstsuche



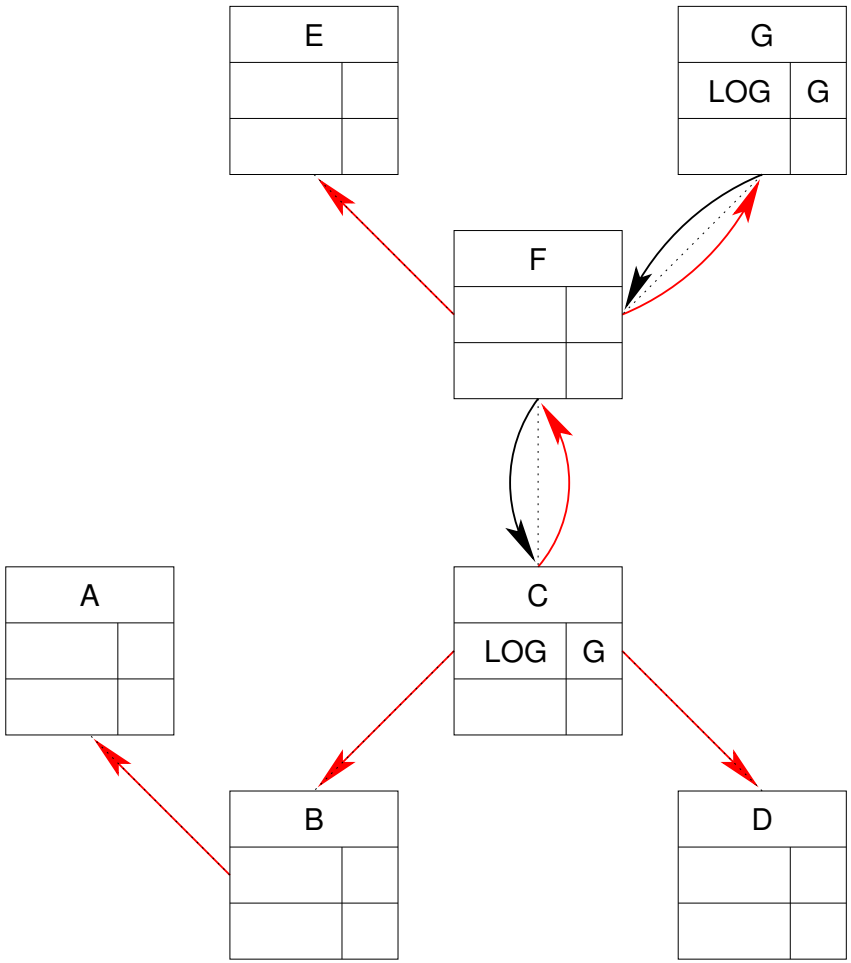
- C sucht Dienst LOG
- flute Anfrage REQ(LOG)
- G beantwortet Anfrage

Dienstsuche



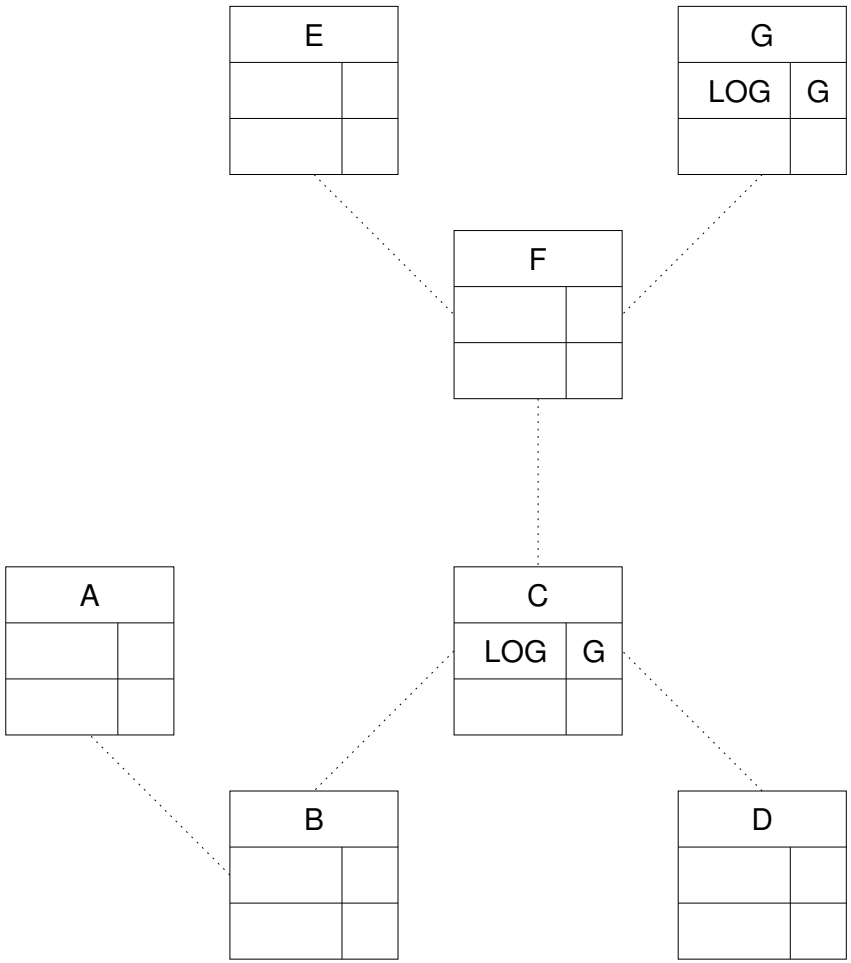
- C sucht Dienst LOG
- flute Anfrage REQ(LOG)
- G beantwortet Anfrage
- Antwort weiterleiten

Dienstsuche



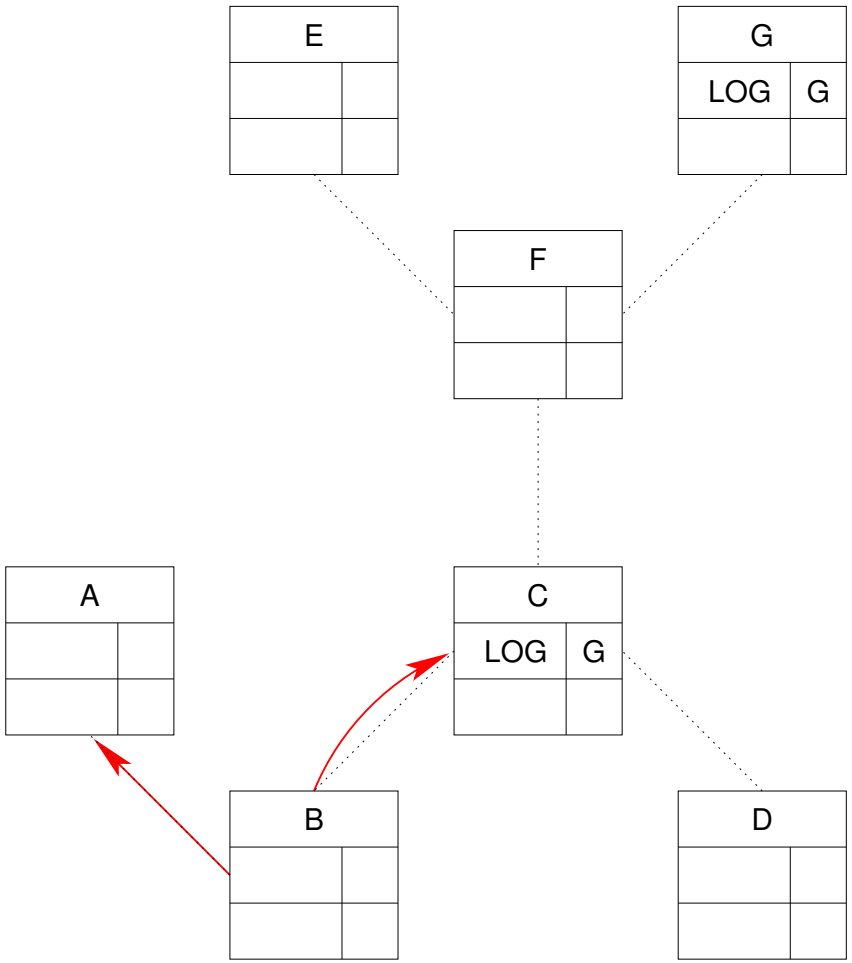
- C sucht Dienst LOG
- flute Anfrage REQ(LOG)
- G beantwortet Anfrage
- Antwort weiterleiten
- C lernt: G bietet Dienst LOG

Dienstsuche



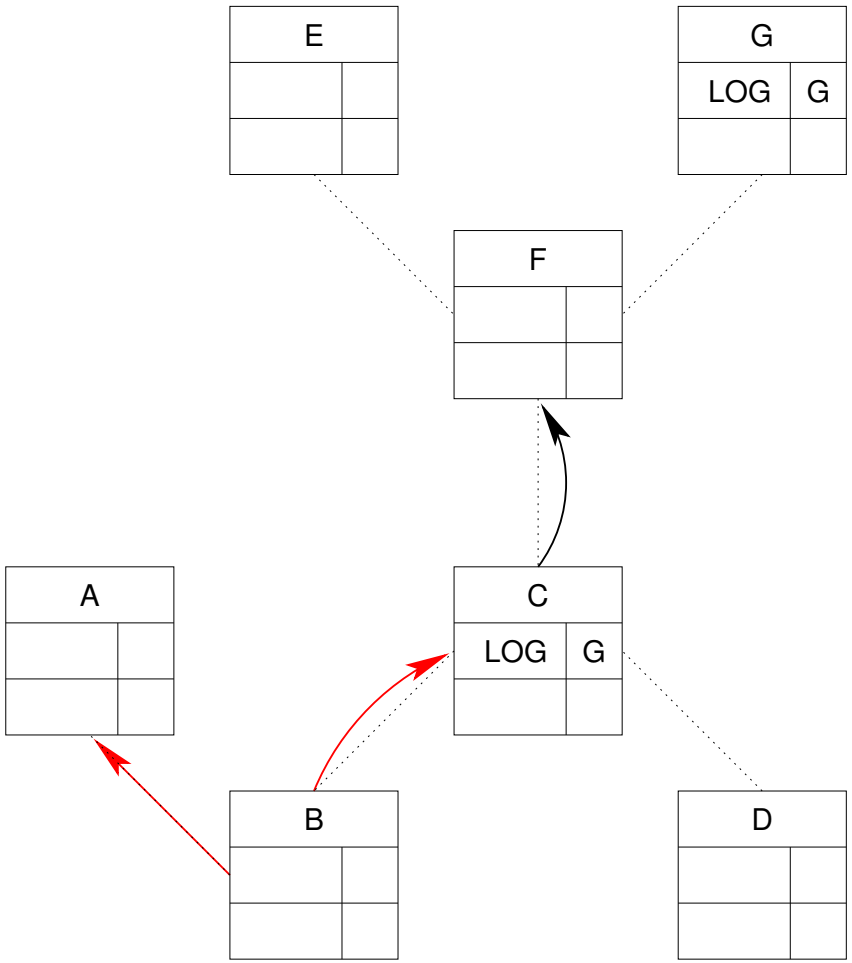
- B sucht Dienst LOG

Dienstsuche



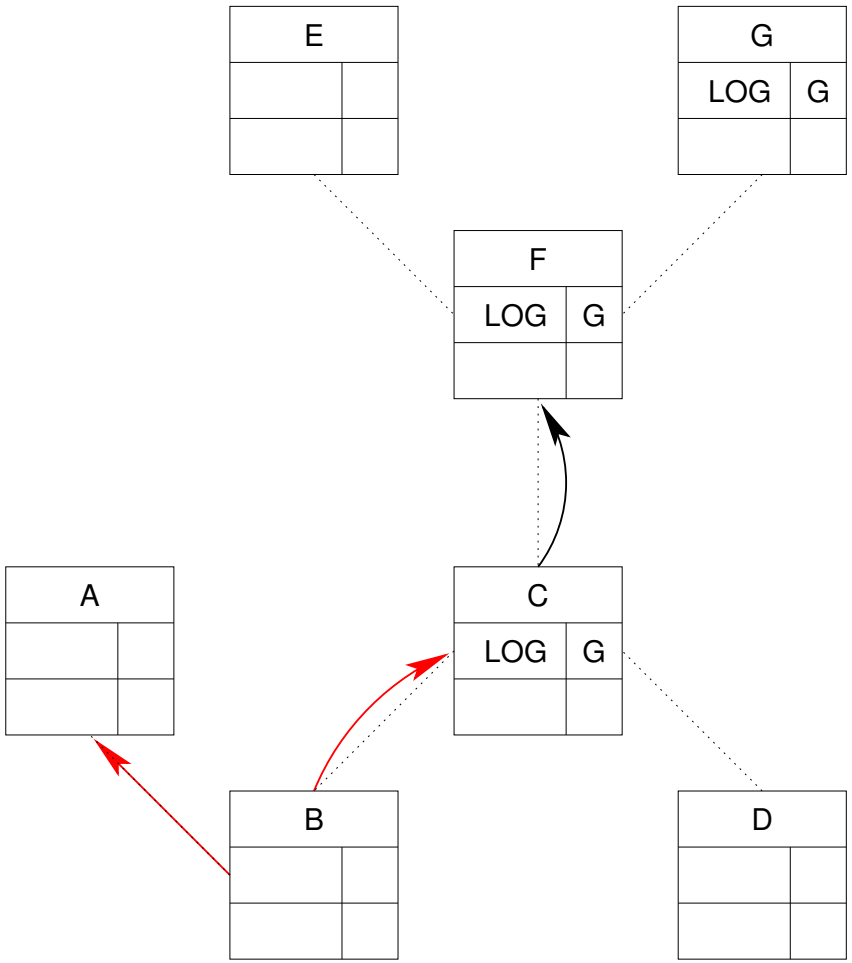
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)

Dienstsuche



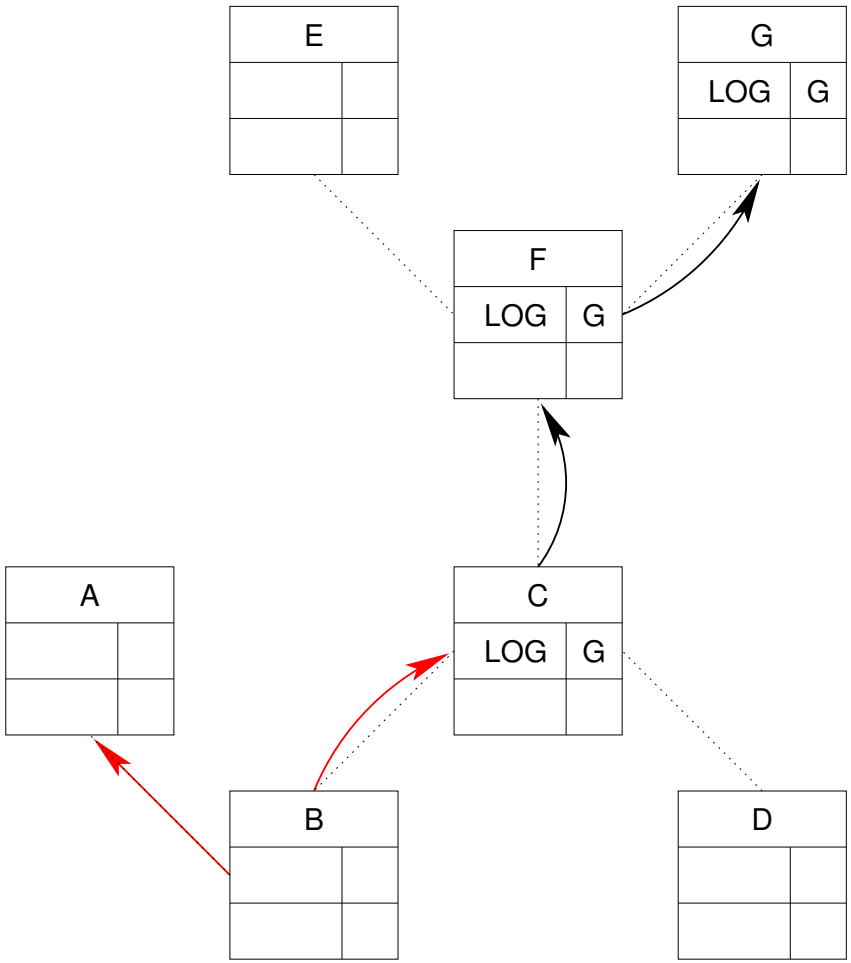
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)

Dienstsuche



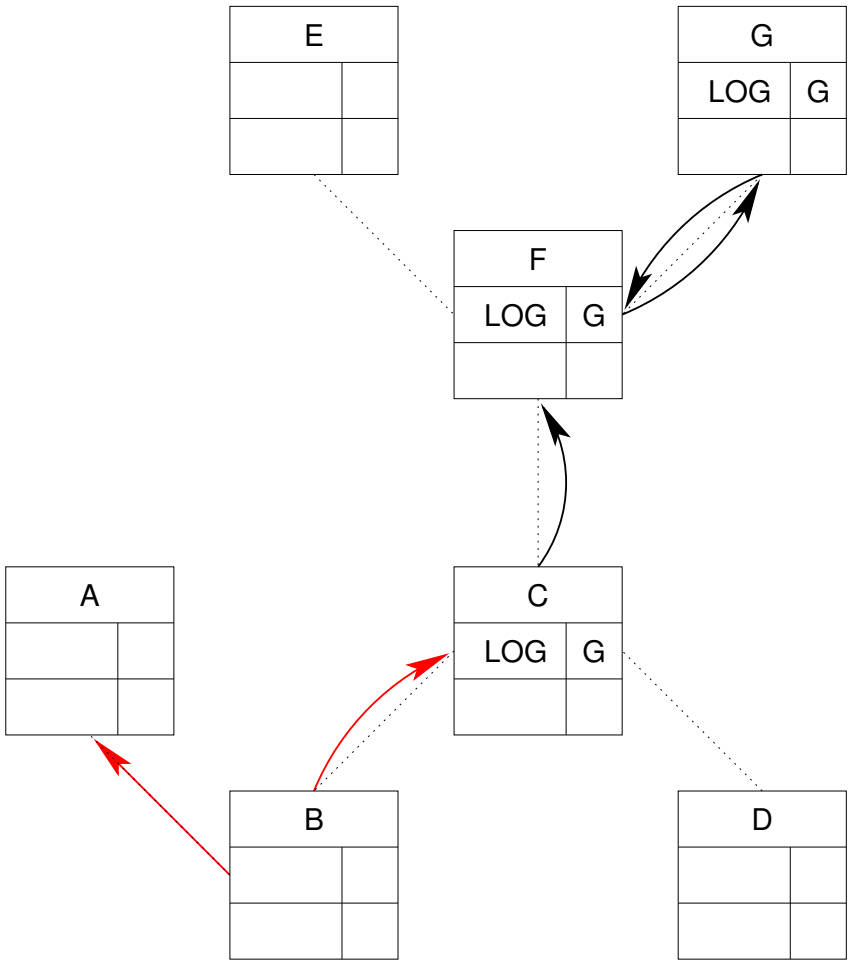
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG

Dienstsuche



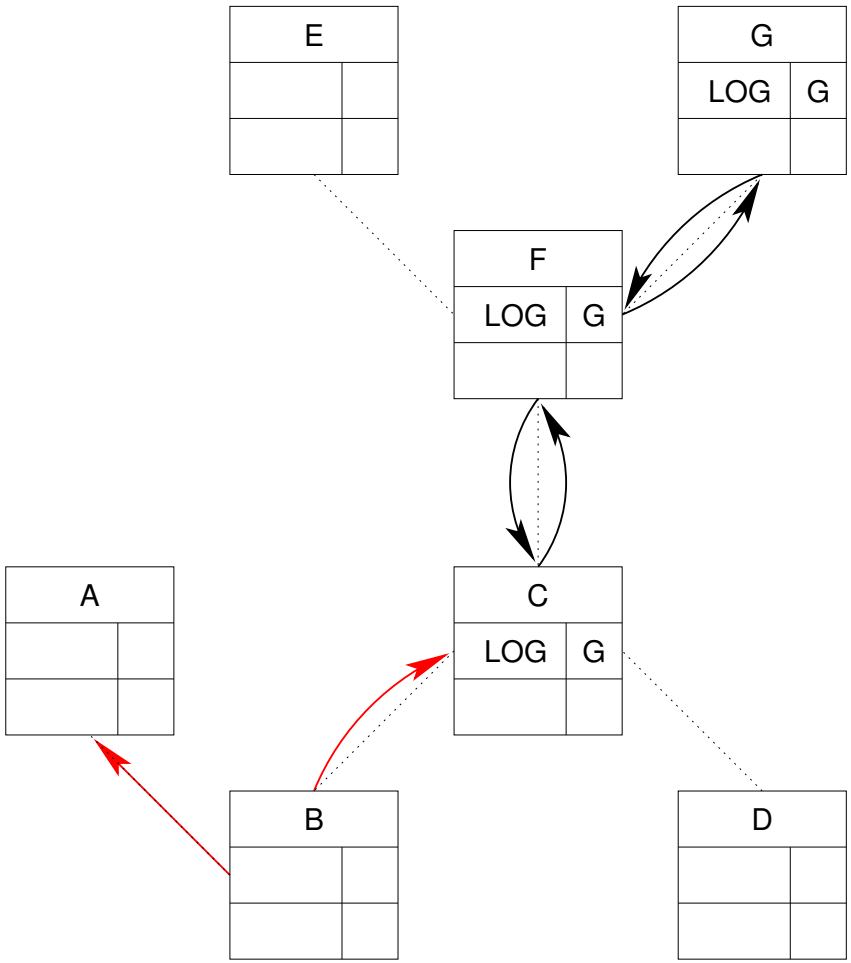
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG

Dienstsuche



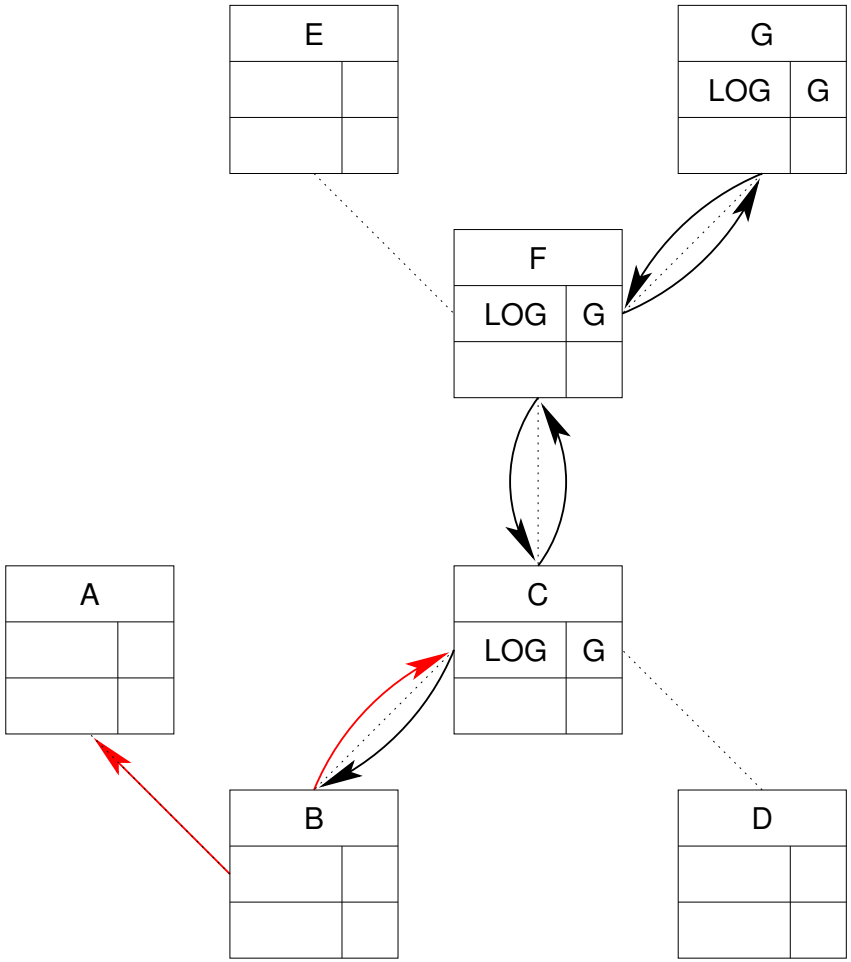
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG
- G beantwortet Anfrage

Dienstsuche



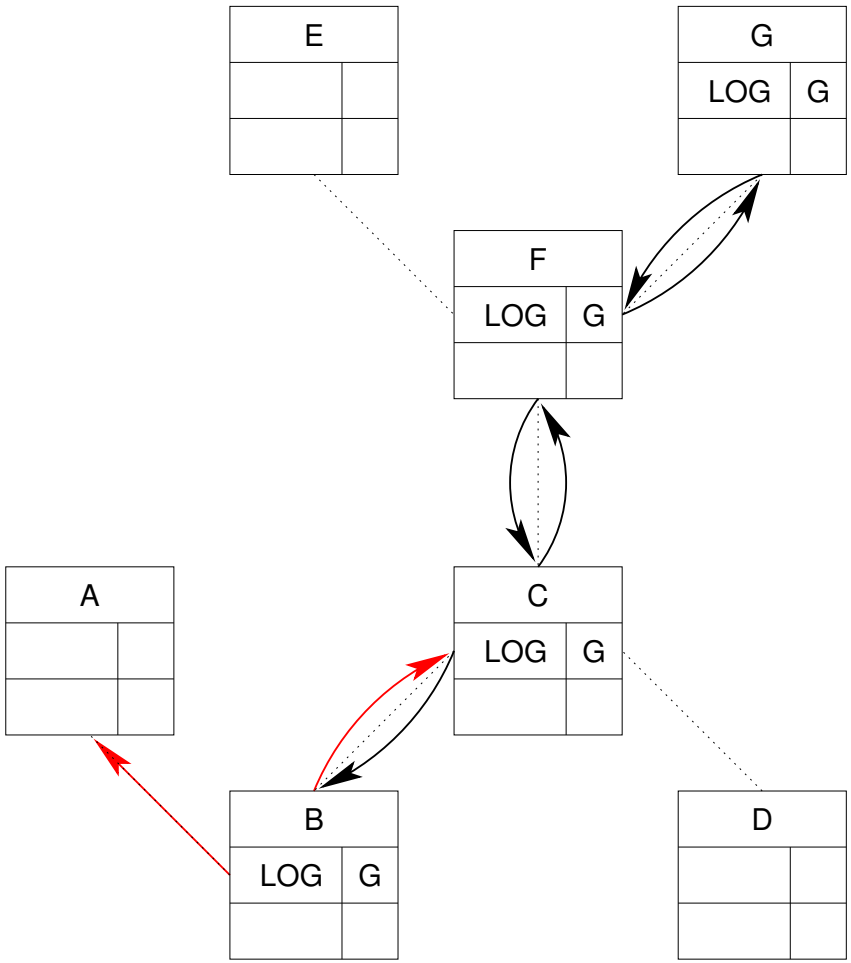
- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG
- G beantwortet Anfrage
- Antwort weiterleiten

Dienstsuche



- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG
- G beantwortet Anfrage
- Antwort weiterleiten

Dienstsuche

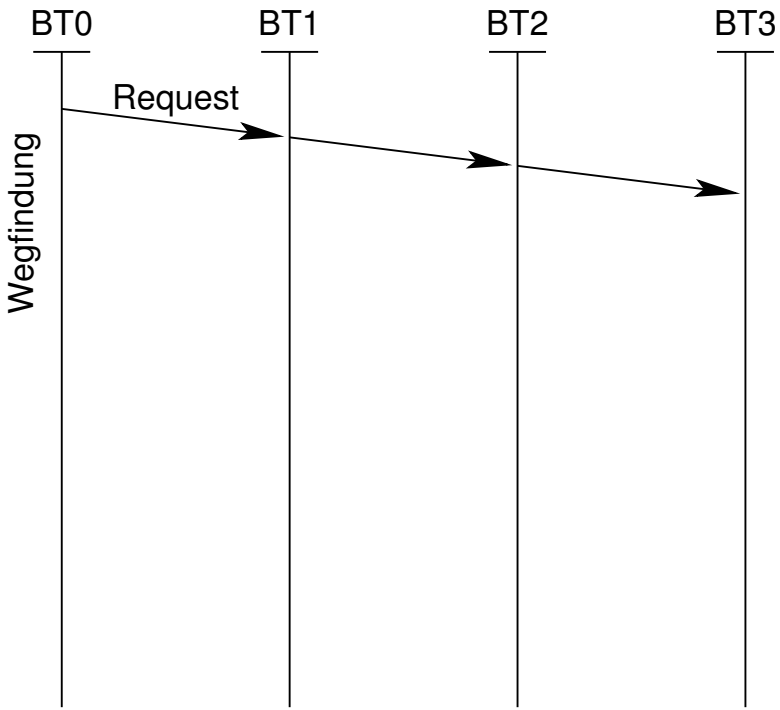


- B sucht Dienst LOG
- flute Anfrage REQ(LOG)
- C löst Anfrage auf REQ(LOG) ⇒ REQ(LOG,G)
- Unicast REQ(LOG,G)
- F lernt: G bietet Dienst LOG
- G beantwortet Anfrage
- Antwort weiterleiten
- B lernt: G bietet Dienst LOG

Datenkanalaufbau

Funktionale Trennung

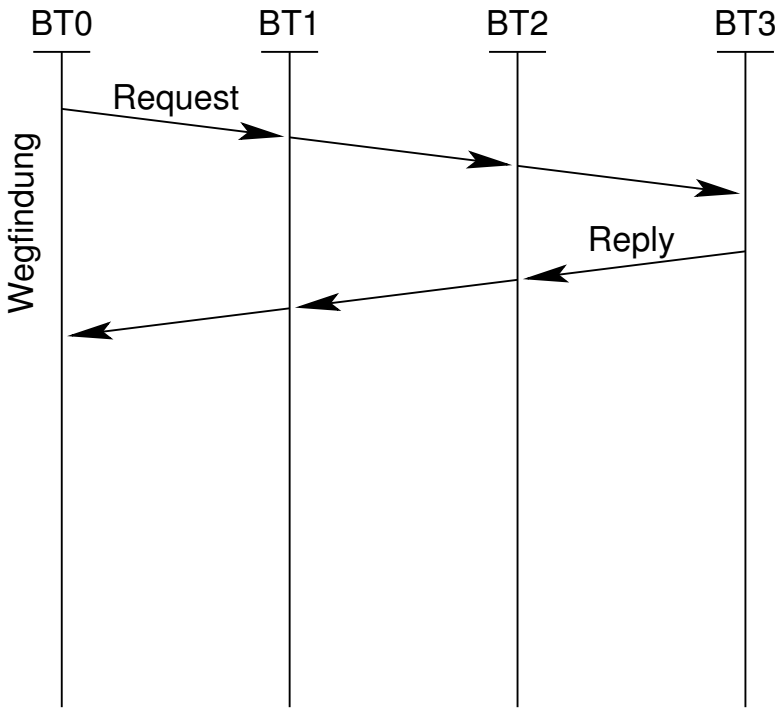
1. Wegfindung



Datenkanalaufbau

Funktionale Trennung

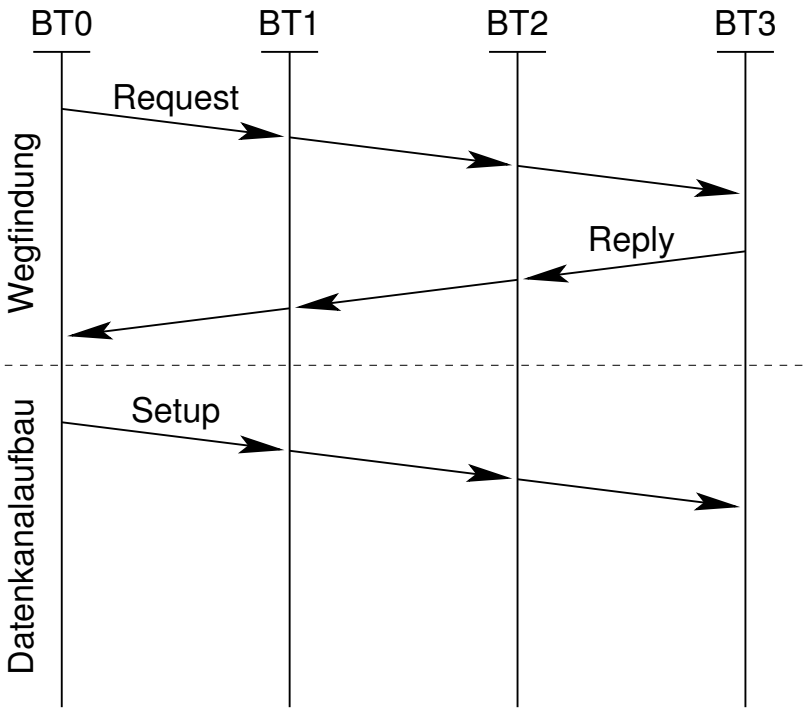
1. Wegfindung (2 Pakete)



Datenkanalaufbau

Funktionale Trennung

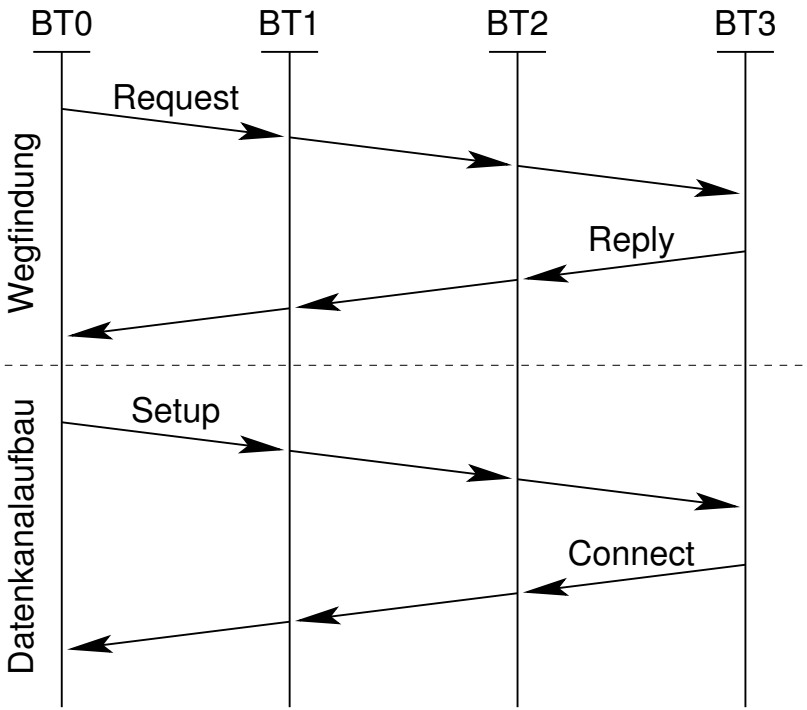
1. Wegfindung (2 Pakete)
2. Datenkanalaufbau



Datenkanalaufbau

Funktionale Trennung

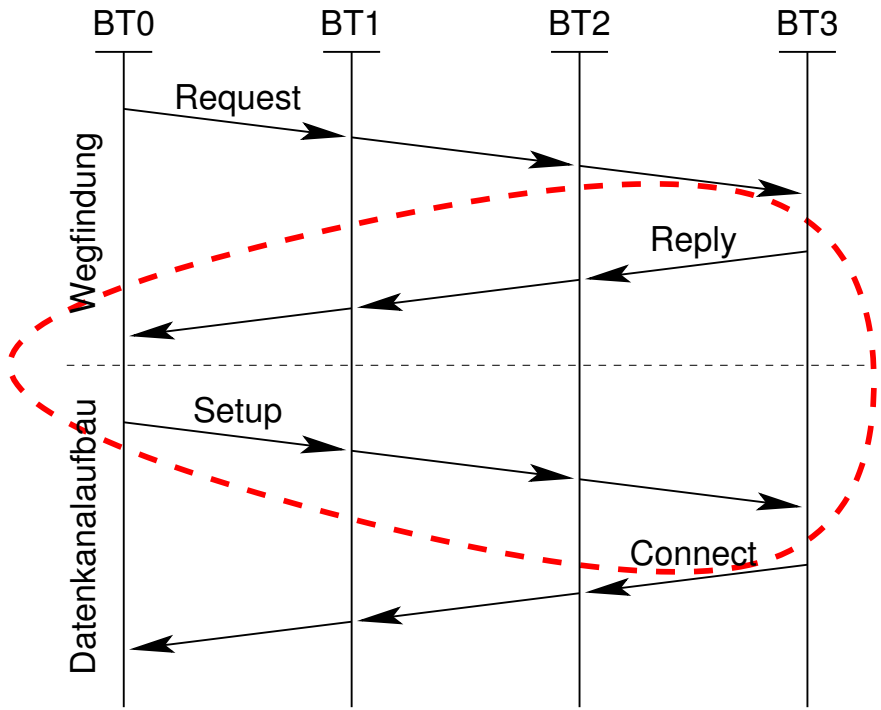
1. Wegfindung (2 Pakete)
2. Datenkanalaufbau (2 Pakete)



Datenkanalaufbau

Funktionale Trennung

1. Wegfindung (2 Pakete)
2. Datenkanalaufbau (2 Pakete)

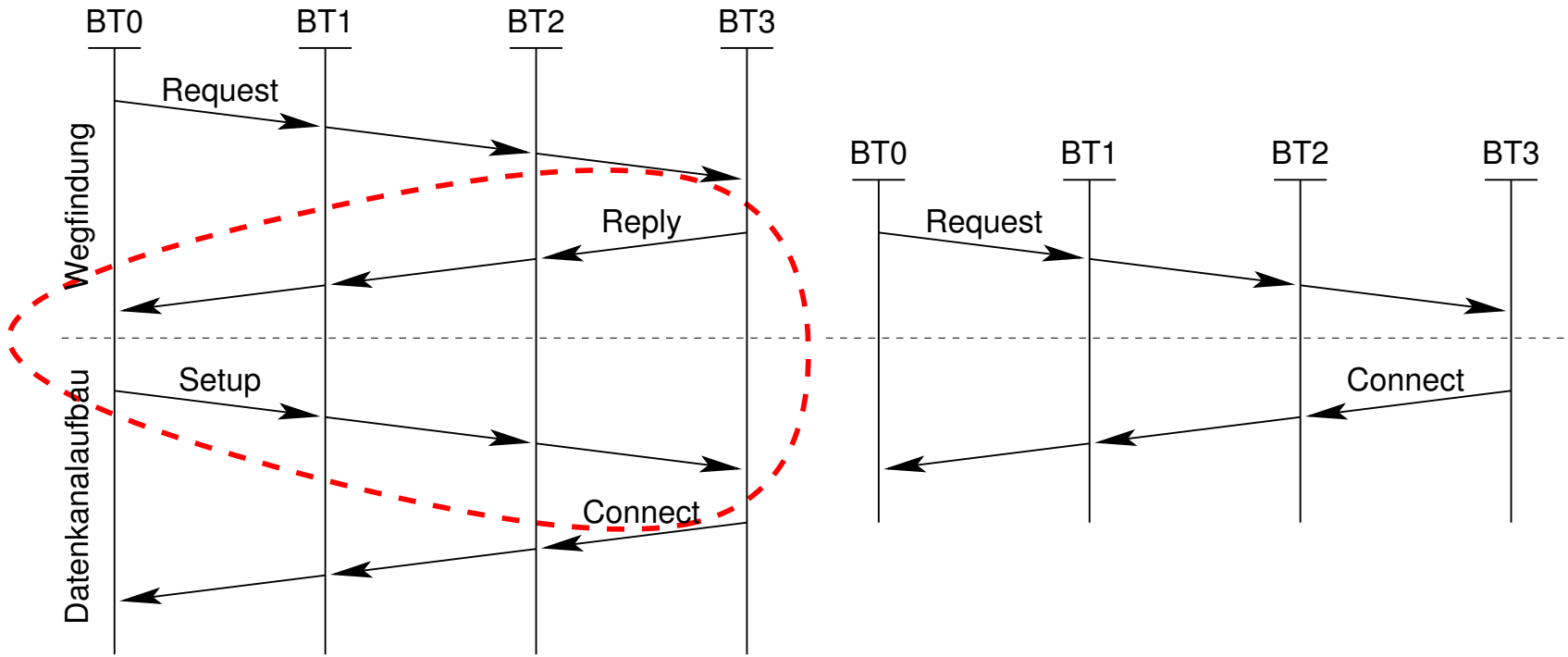


Datenkanalaufbau

Funktionale Trennung

1. Wegfindung (2 Pakete)
2. Datenkanalaufbau (2 Pakete)

Kombiniert



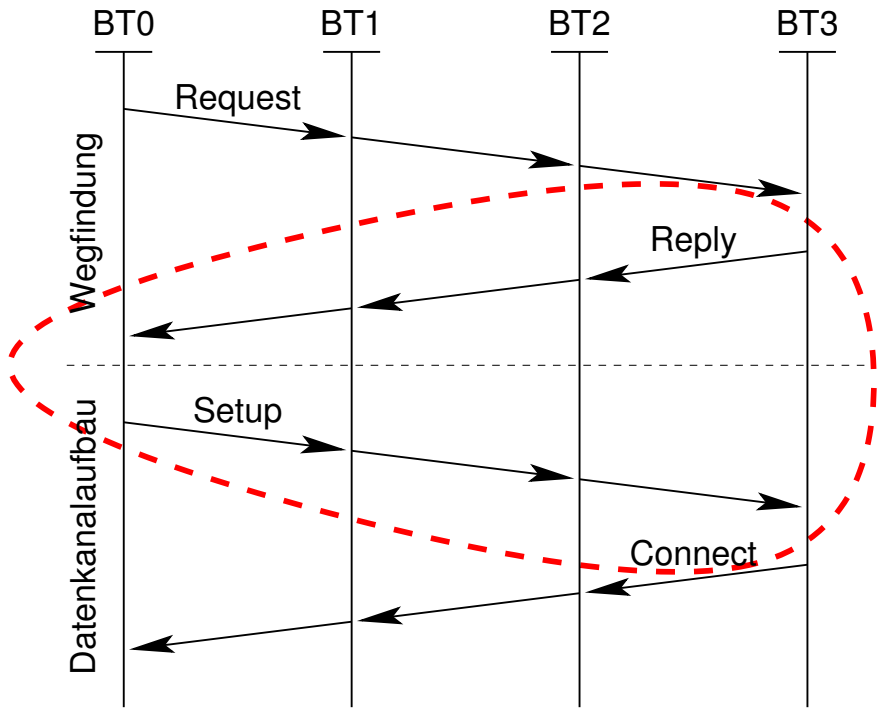
Datenkanalaufbau

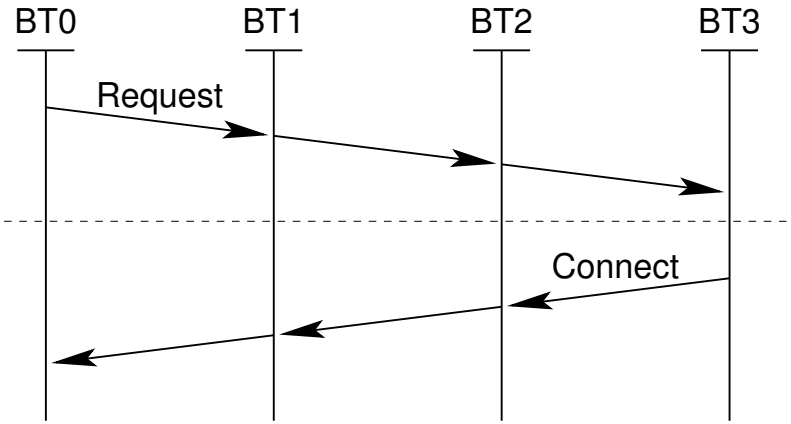
Funktionale Trennung

1. Wegfindung (2 Pakete)
2. Datenkanalaufbau (2 Pakete)

Kombiniert

1. Hinweg: Topologieausprägung
   Wegfindung
   Dienstsuche





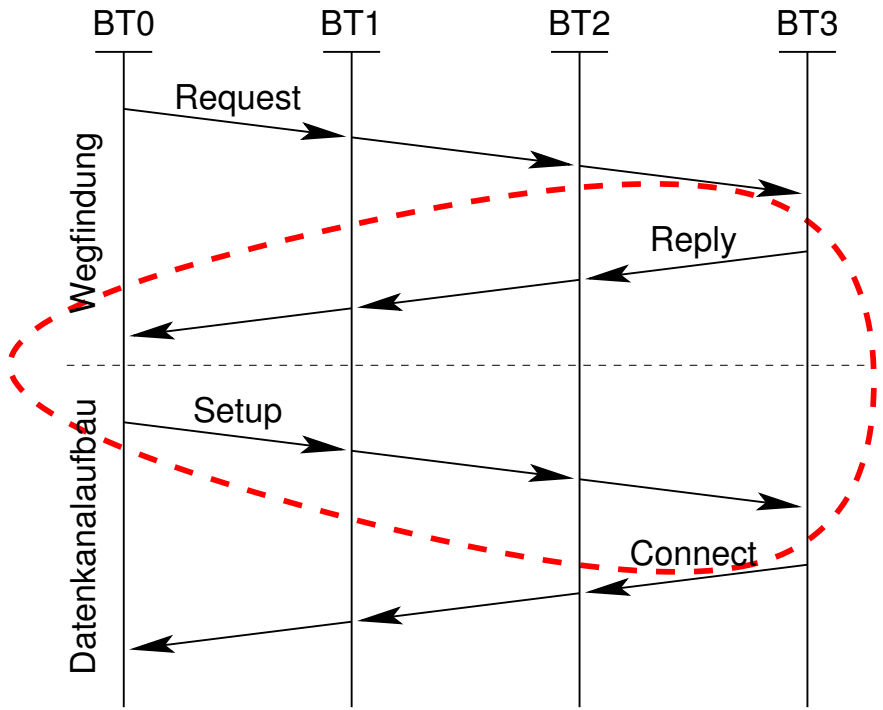
Datenkanalaufbau

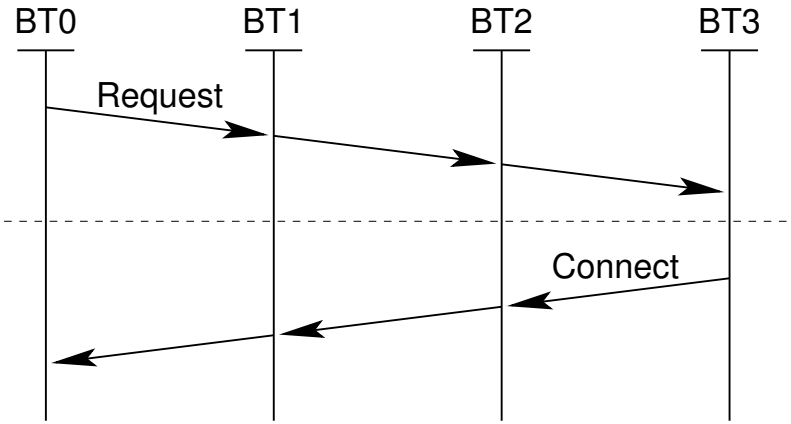
Funktionale Trennung

1. Wegfindung (2 Pakete)
2. Datenkanalaufbau (2 Pakete)

Kombiniert

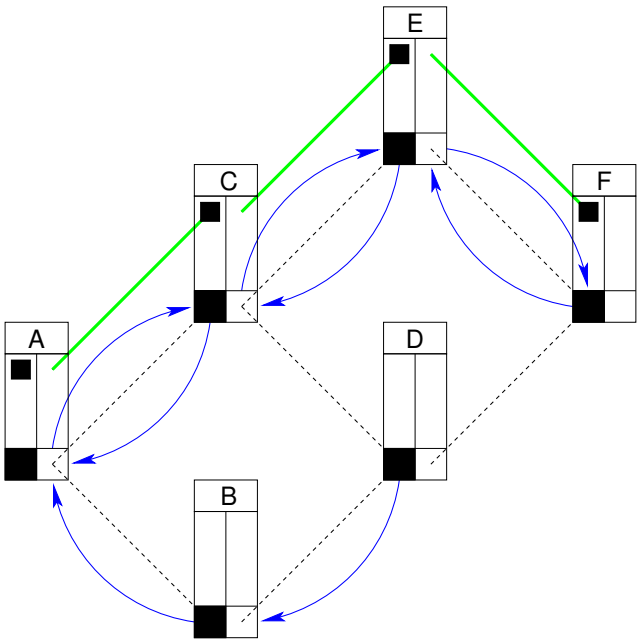
1. Hinweg: Topologieausprägung
   Wegfindung
   Dienstsuche
2. Rückweg: Datenkanalaufbau



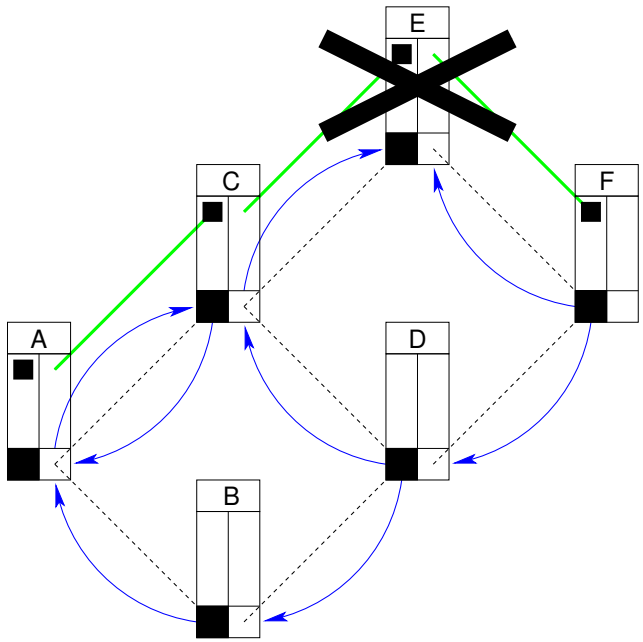


SNR Beispielszenarien

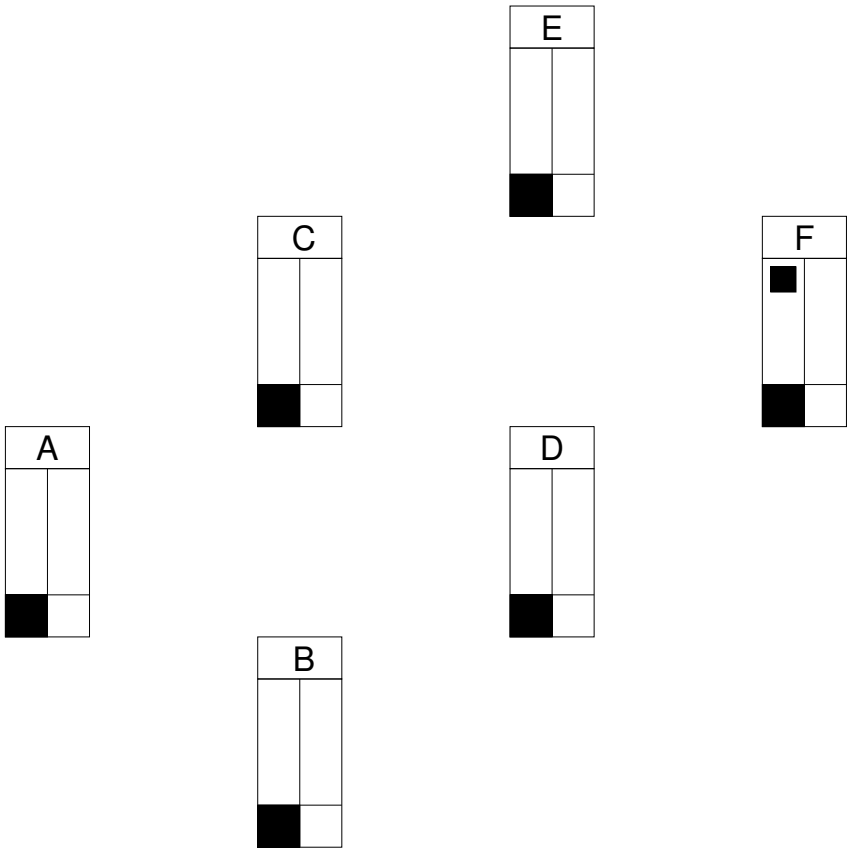
Verbindungsaufbau



Verbindungsausfall

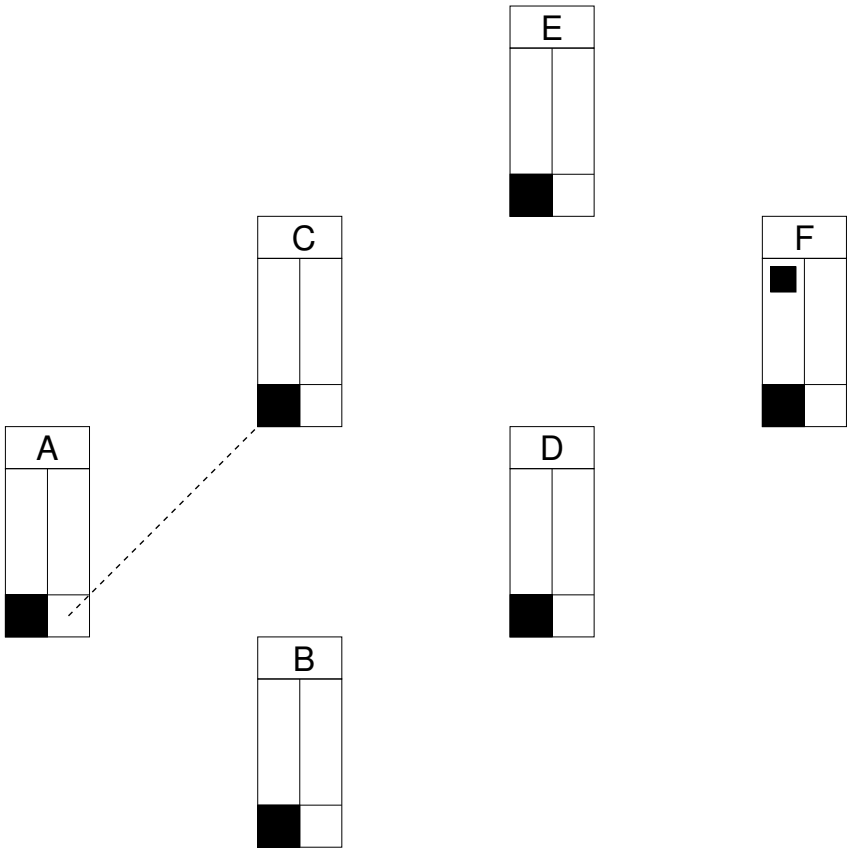


Verbindungsaufbau



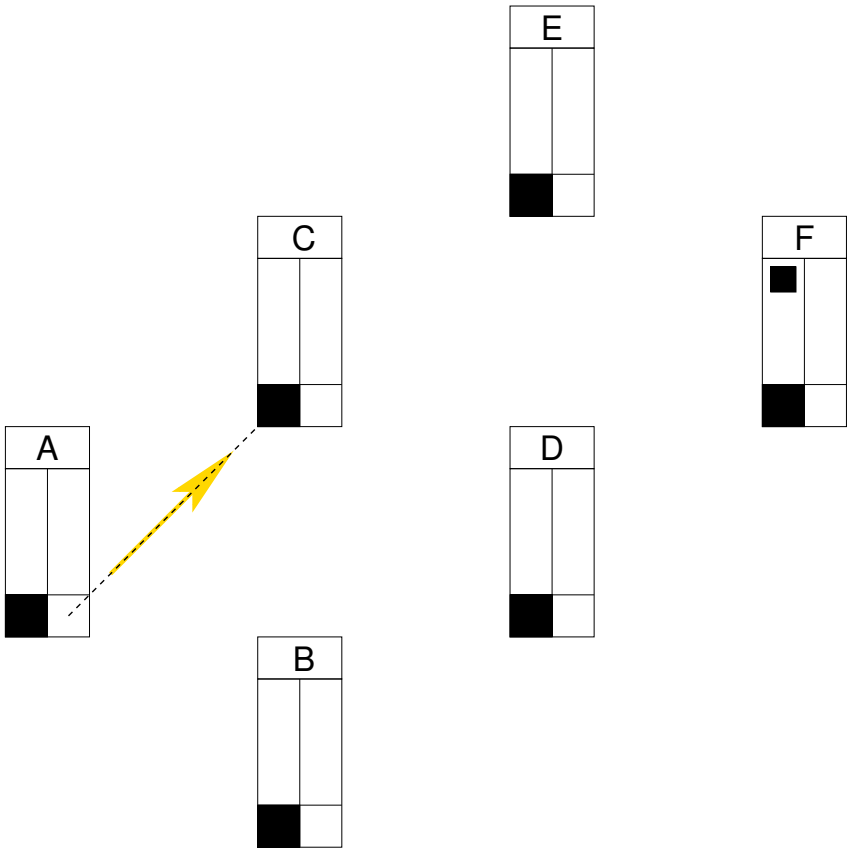
- A benötigt Verbindung zu F

Verbindungsaufbau



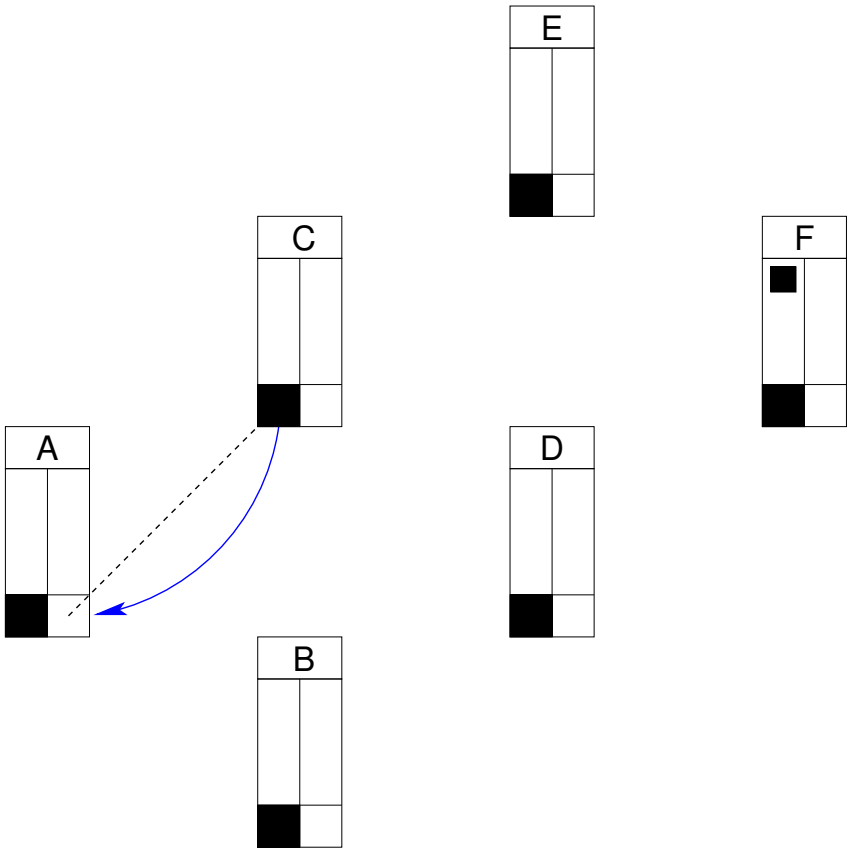
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen

Verbindungsaufbau



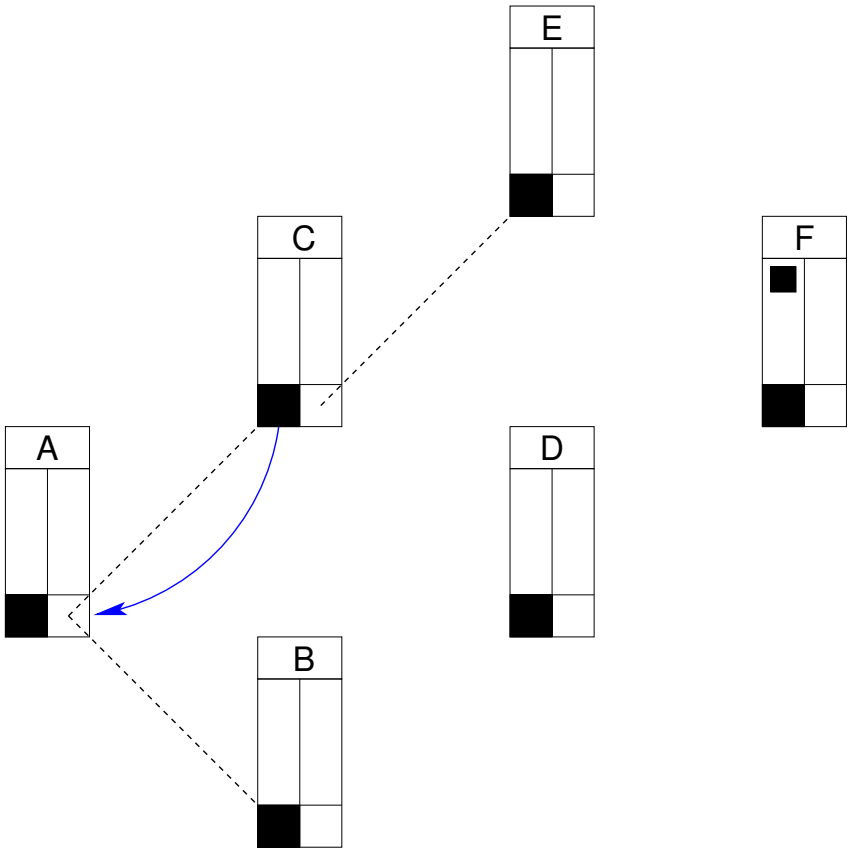
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken

Verbindungsaufbau



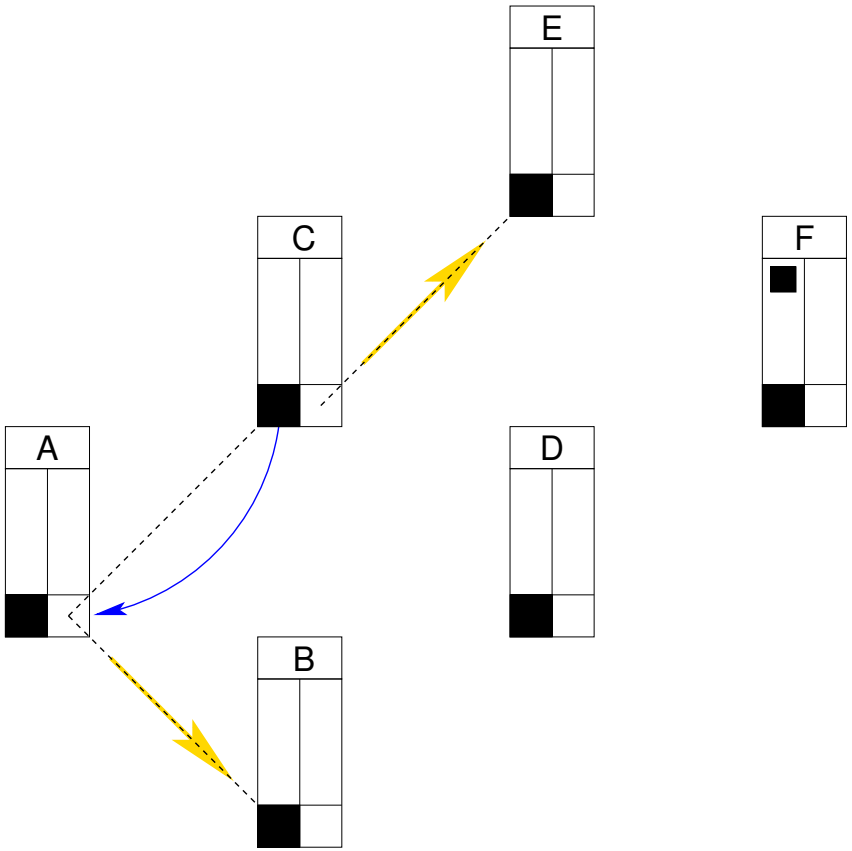
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen

Verbindungsaufbau



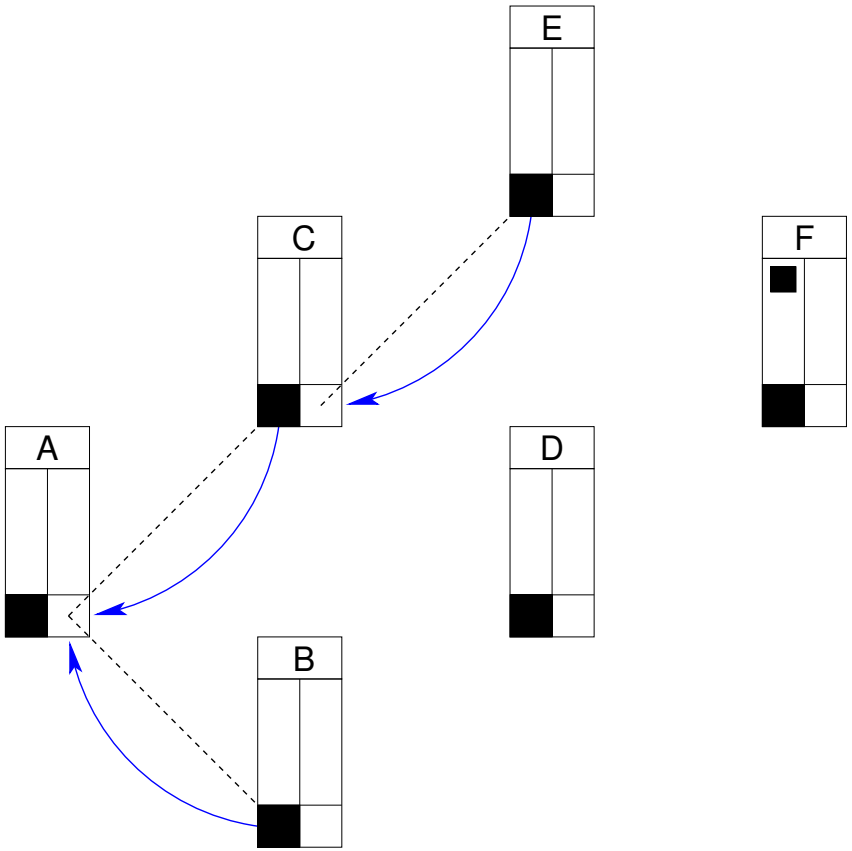
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen

Verbindungsaufbau



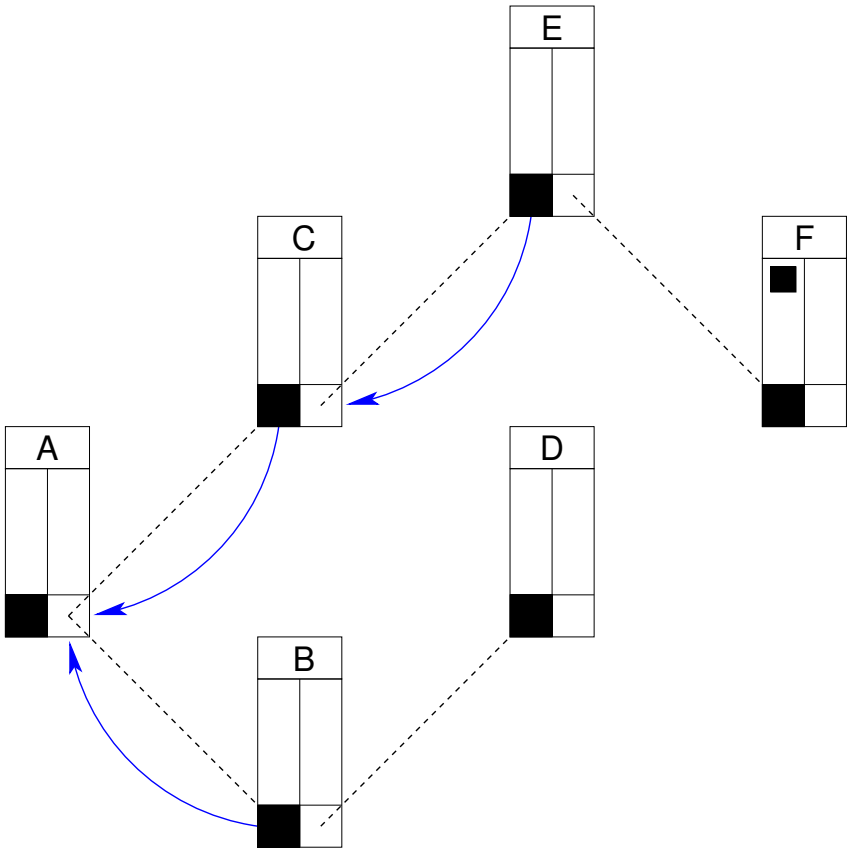
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen

Verbindungsaufbau



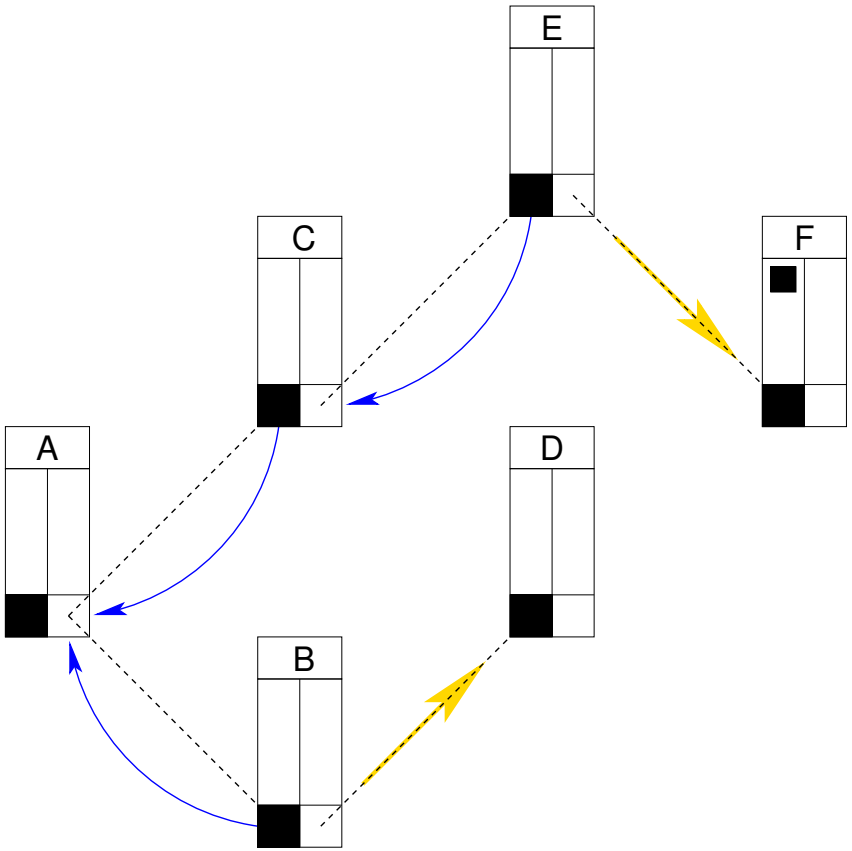
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen

Verbindungsaufbau



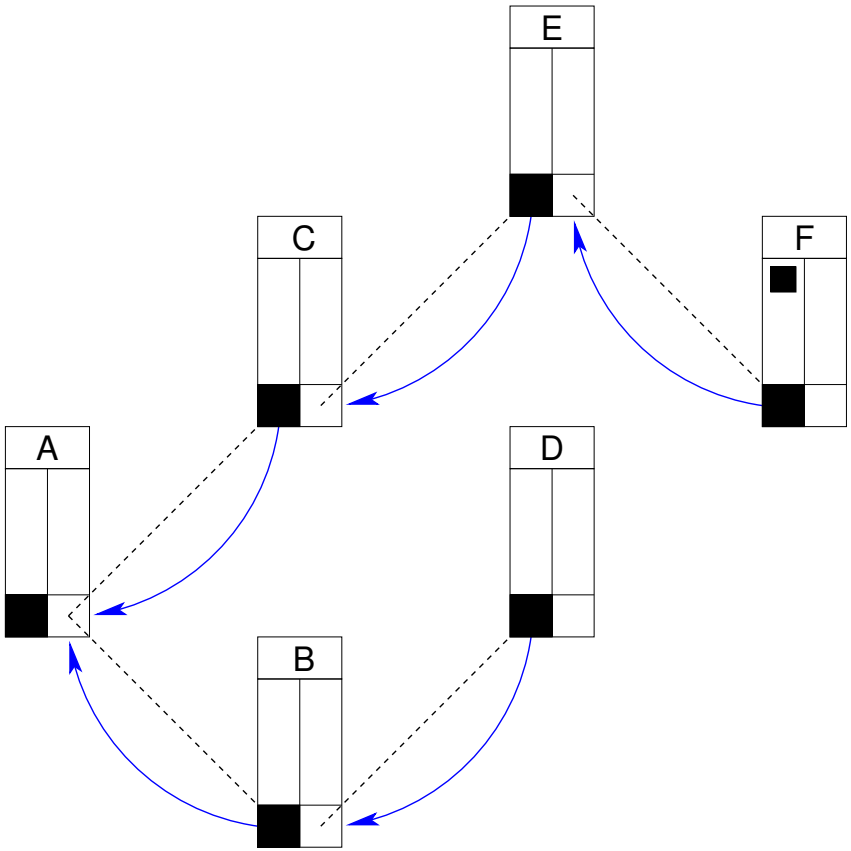
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen

Verbindungsaufbau



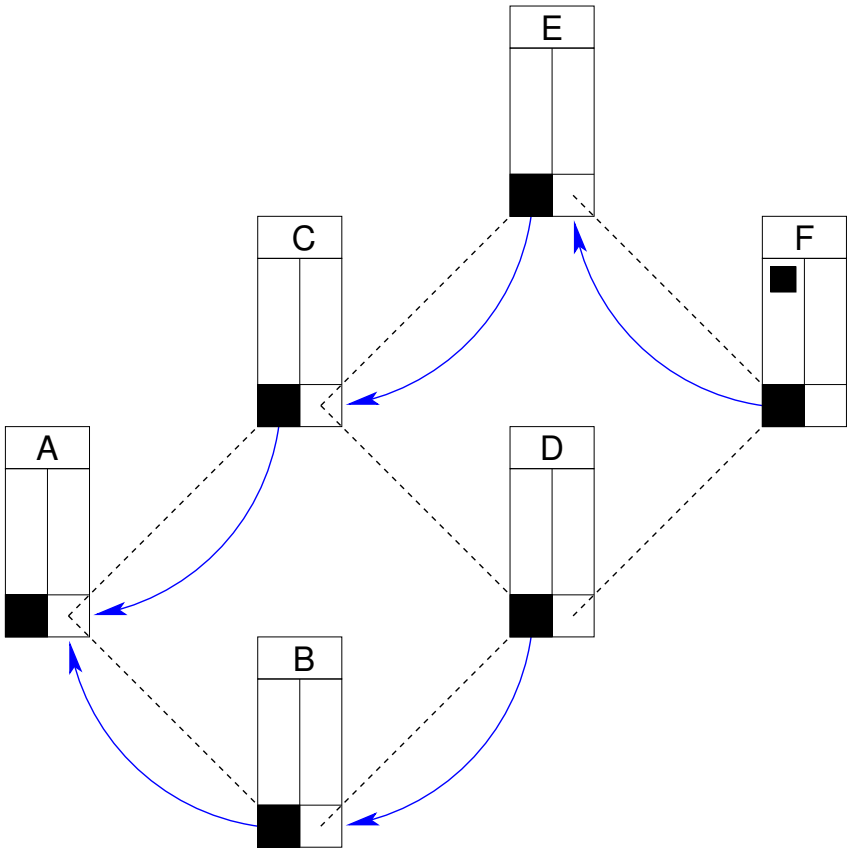
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen
- Anfrage hat Zielknoten erreicht

Verbindungsaufbau



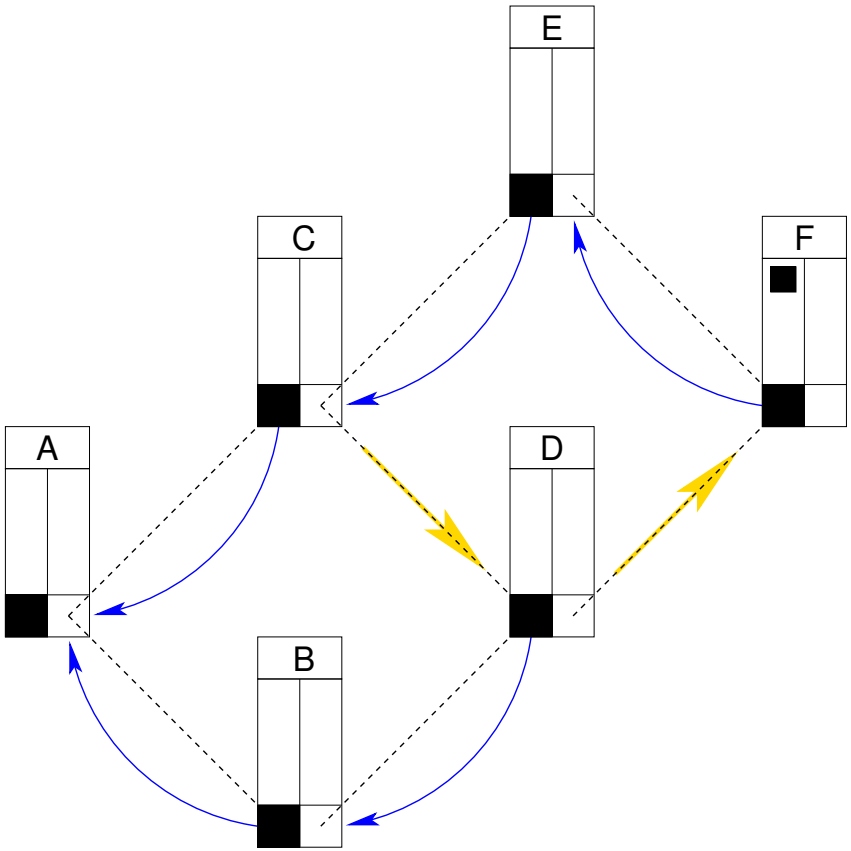
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen
- Anfrage hat Zielknoten erreicht

Verbindungsaufbau



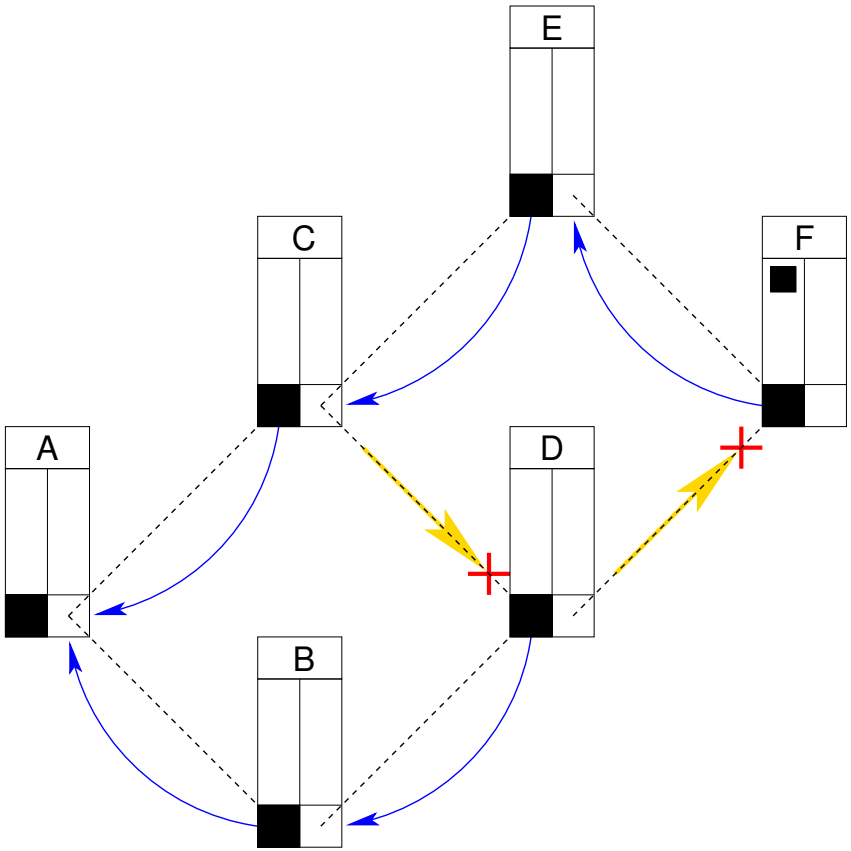
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen
- Anfrage hat Zielknoten erreicht

Verbindungsaufbau



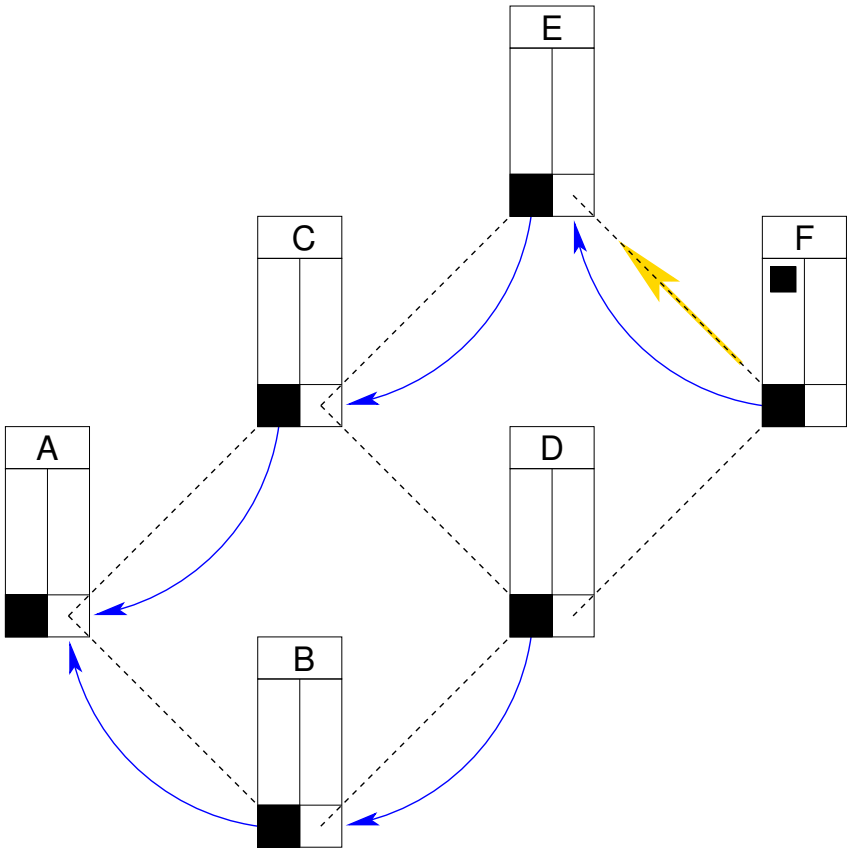
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen
- Anfrage hat Zielknoten erreicht

Verbindungsaufbau



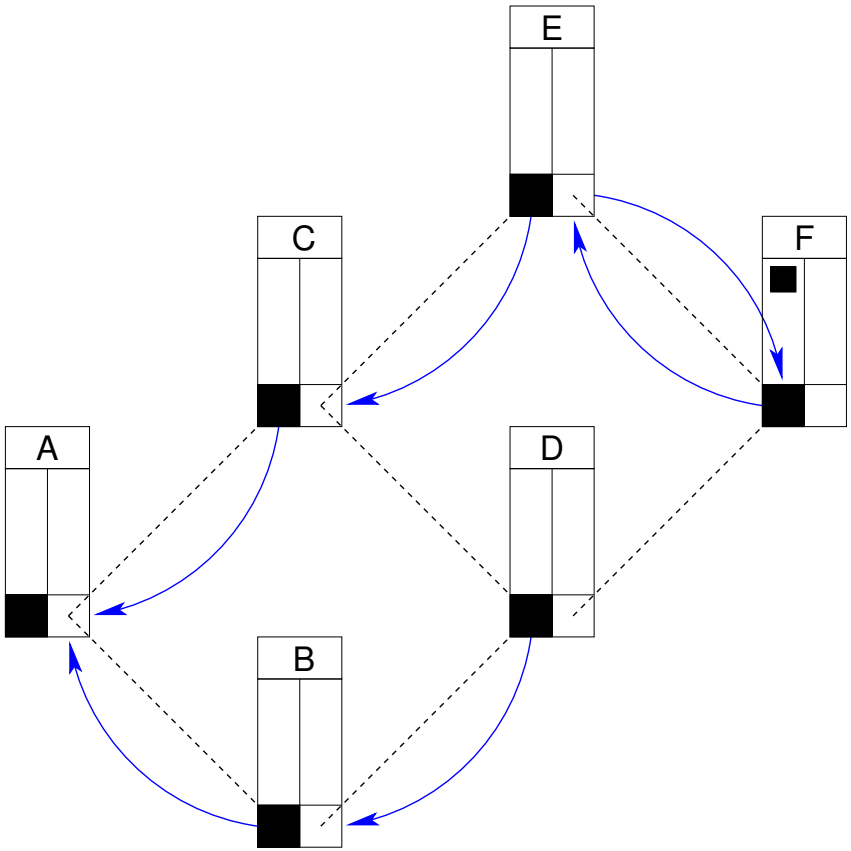
- A benötigt Verbindung zu F
- Signalisierungsverbindung aufbauen
- Anfrage schicken
- Rückweg in Routingtabelle eintragen
- Anfrage hat Zielknoten erreicht
- Anfrage bereits bearbeitet

Verbindungsaufbau



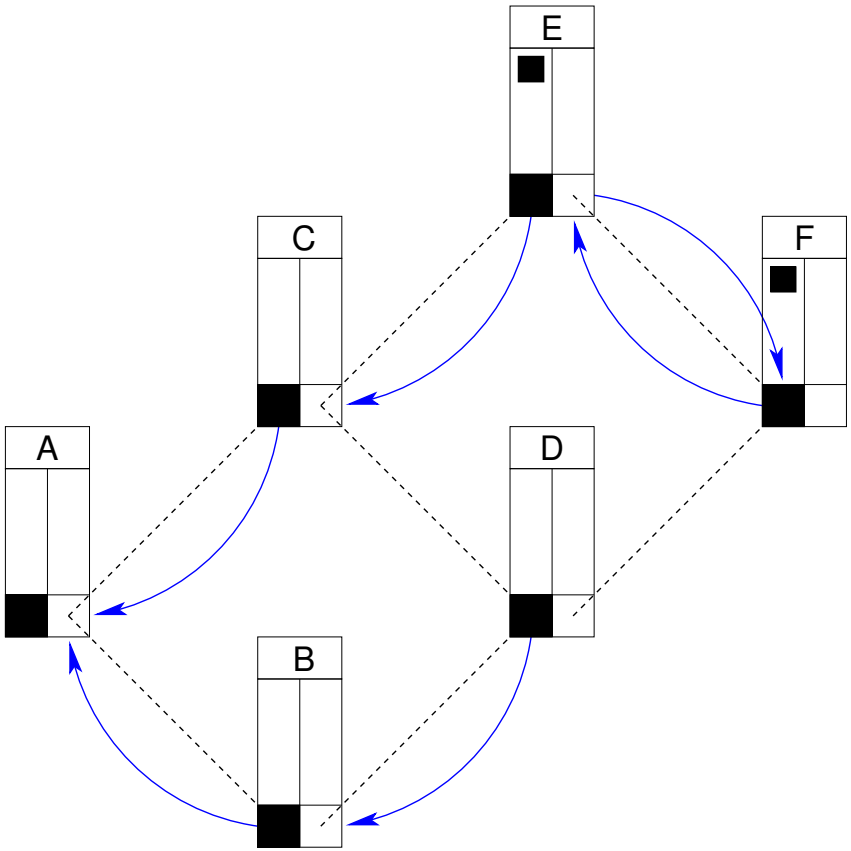
- Antwort schicken

Verbindungsaufbau



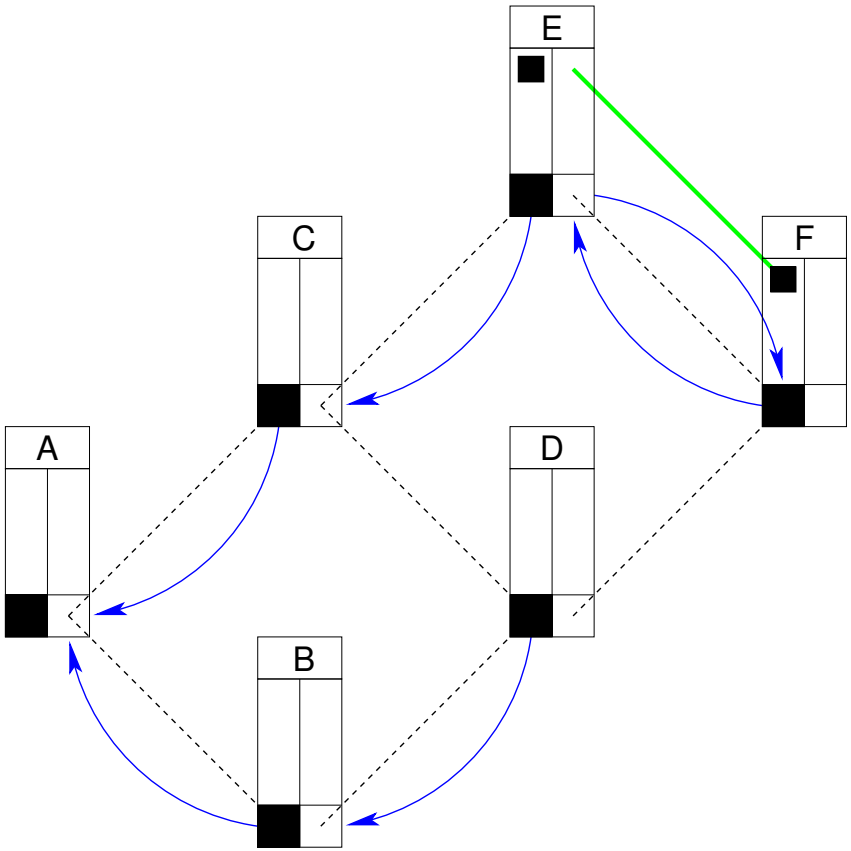
- Antwort schicken
- Hinweg in Routingtabelle eintragen

Verbindungsaufbau



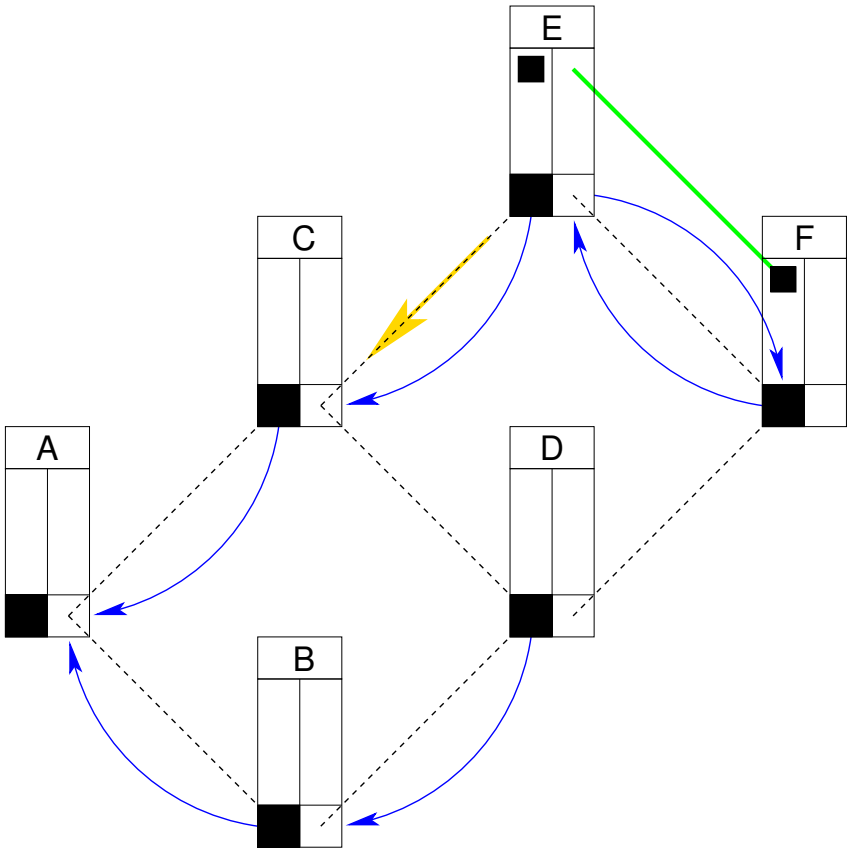
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten

Verbindungsaufbau



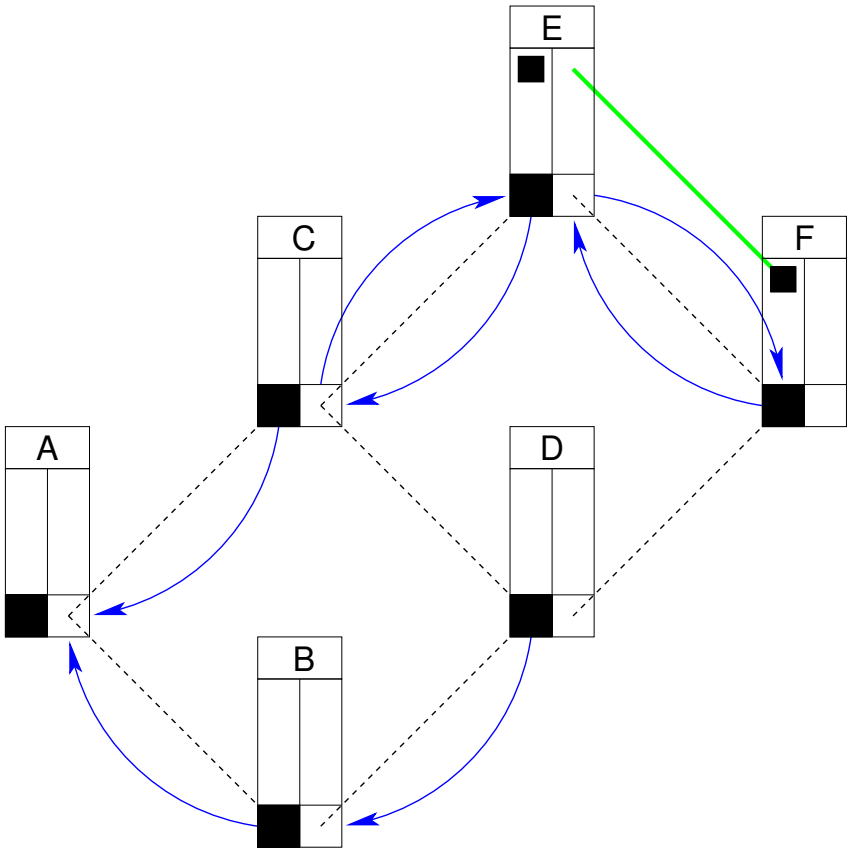
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



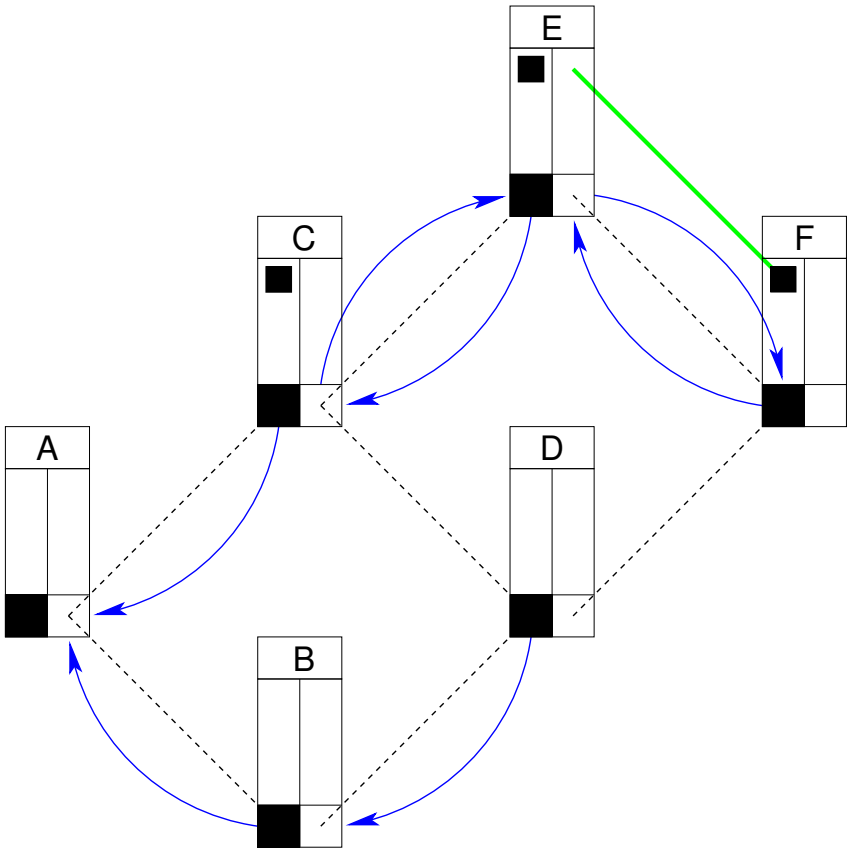
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



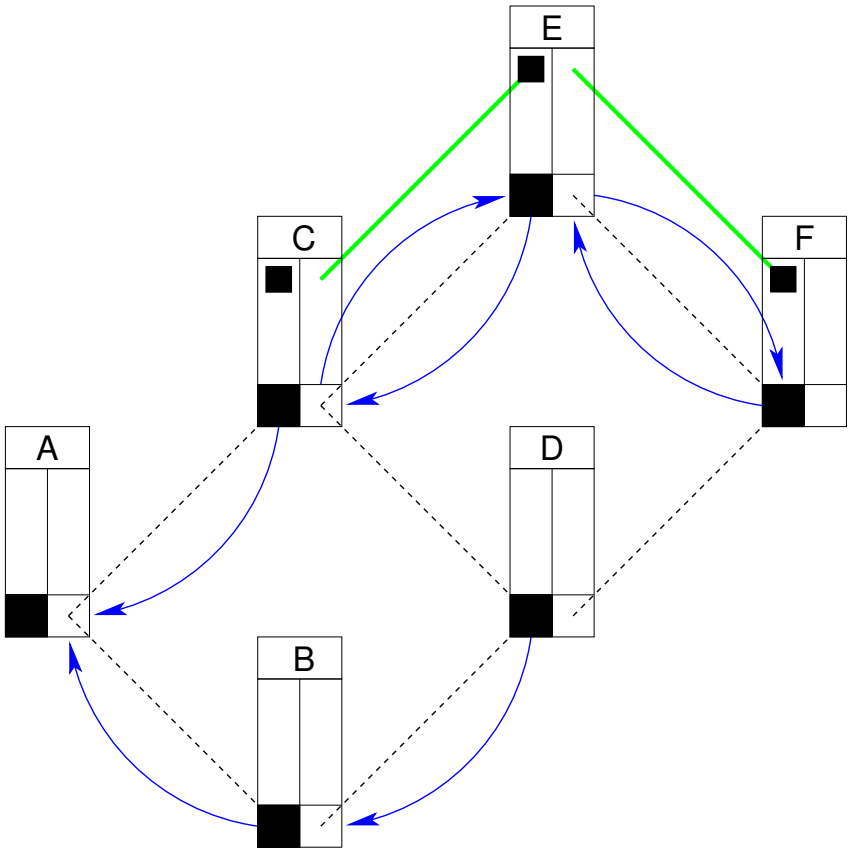
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



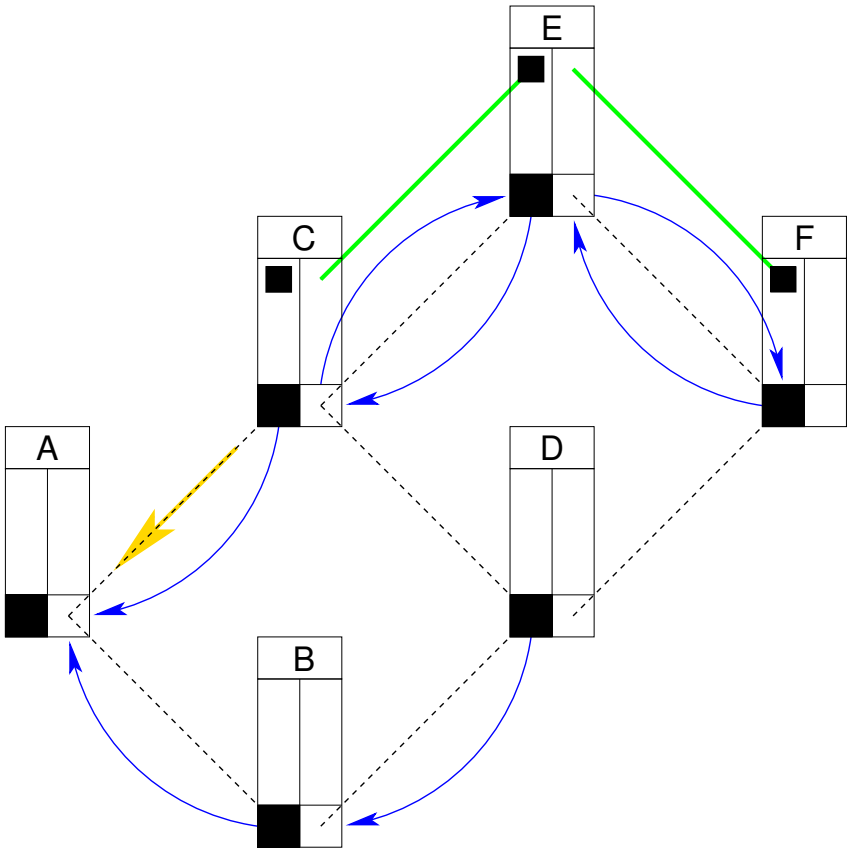
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



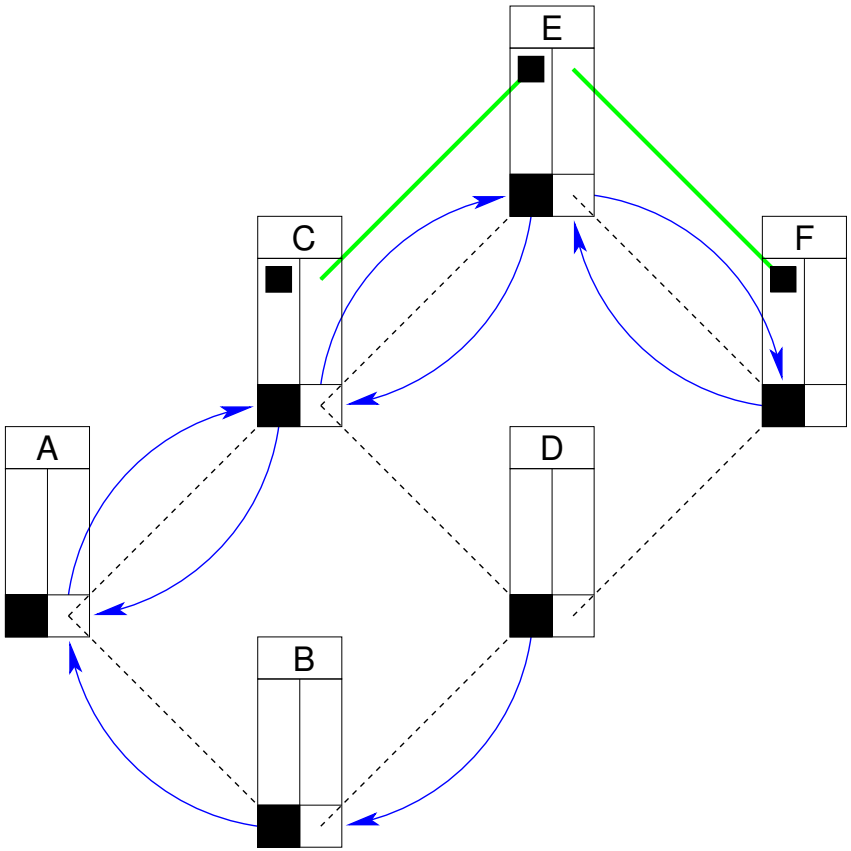
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



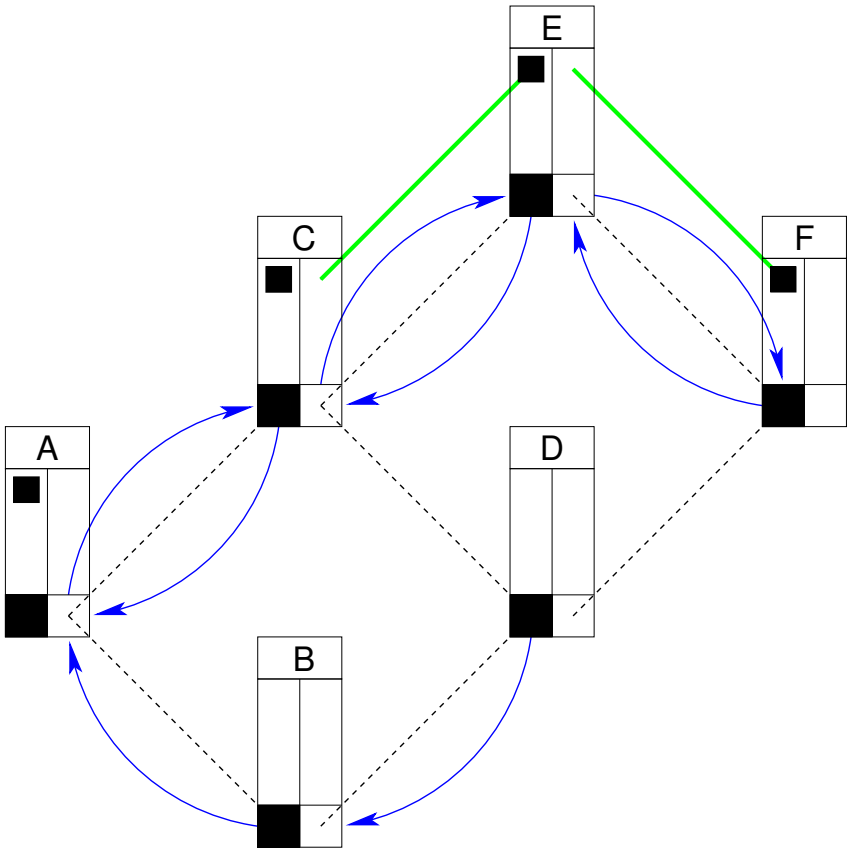
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



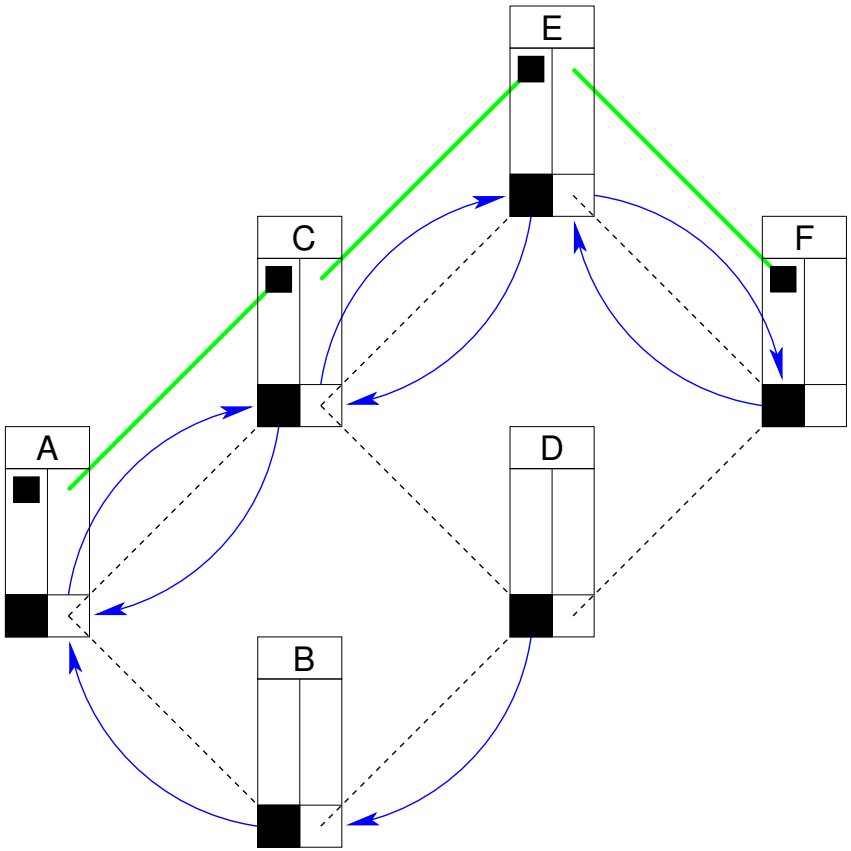
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



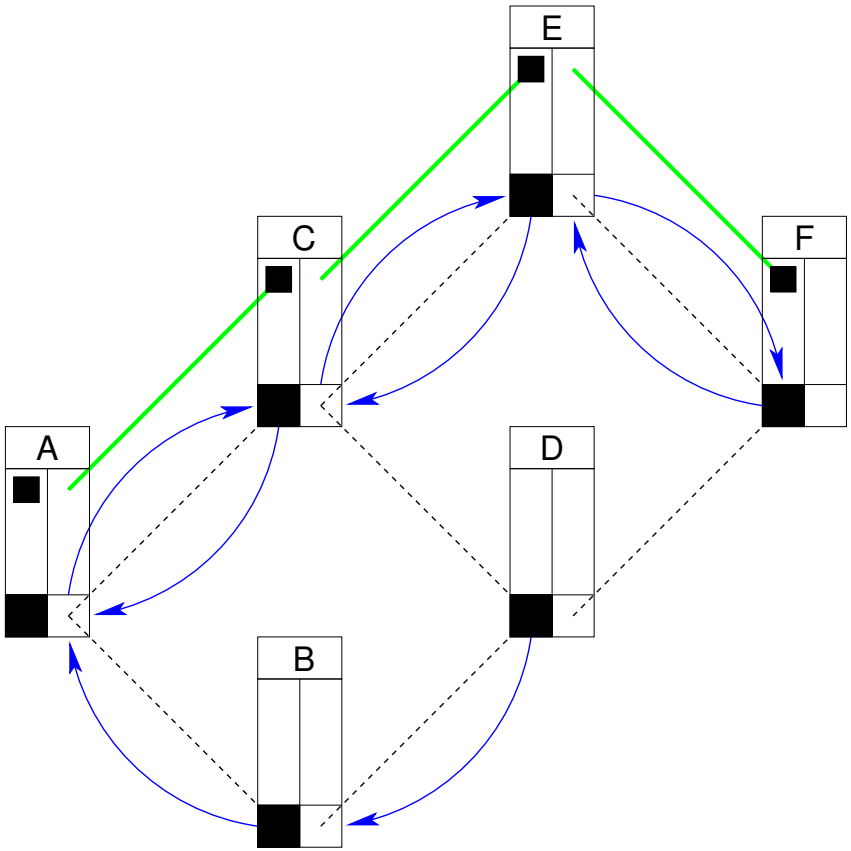
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



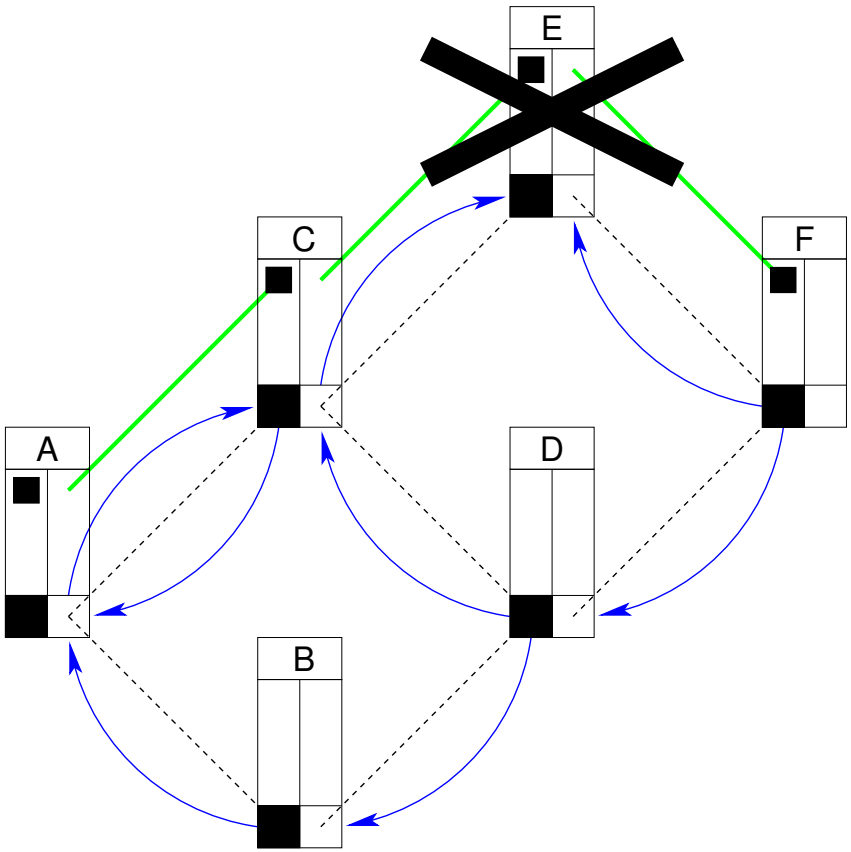
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen

Verbindungsaufbau



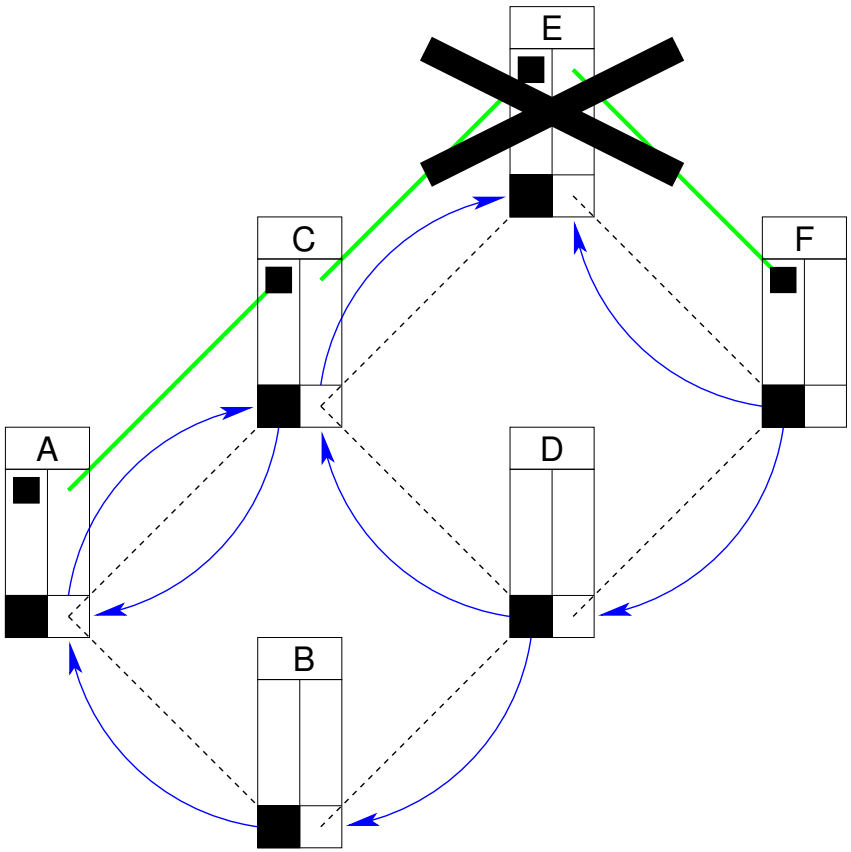
- Antwort schicken
- Hinweg in Routingtabelle eintragen
- Serverdienst starten
- Verbindung zum letzten Knoten aufbauen
- Datenaustausch...

Verbindungsausfall



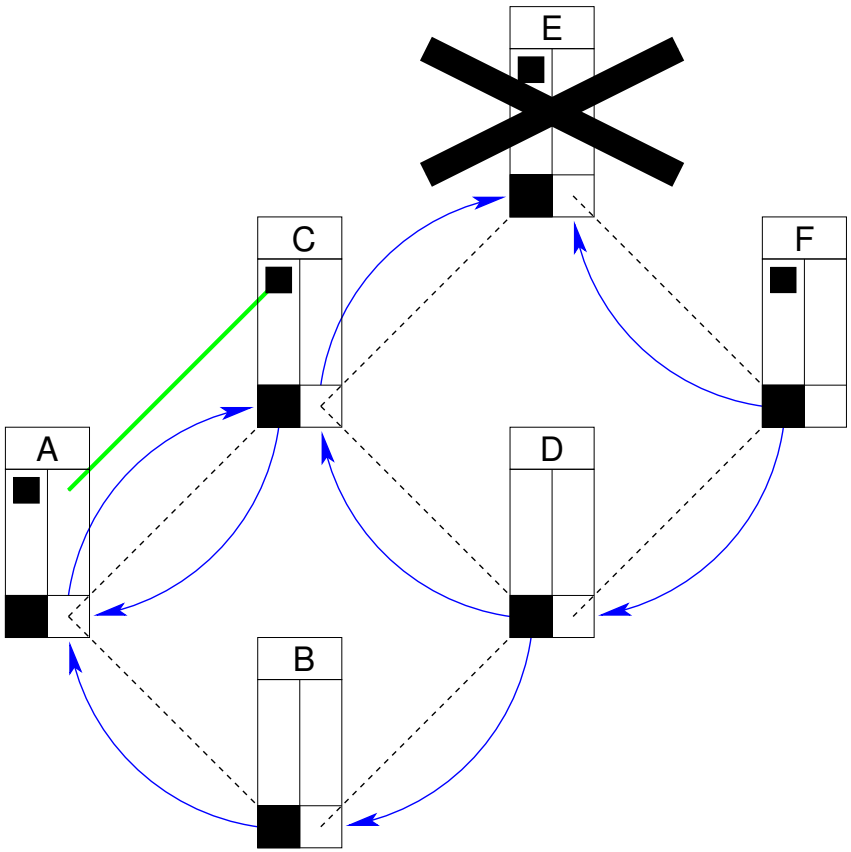
- Knoten E fällt aus

Verbindungsausfall



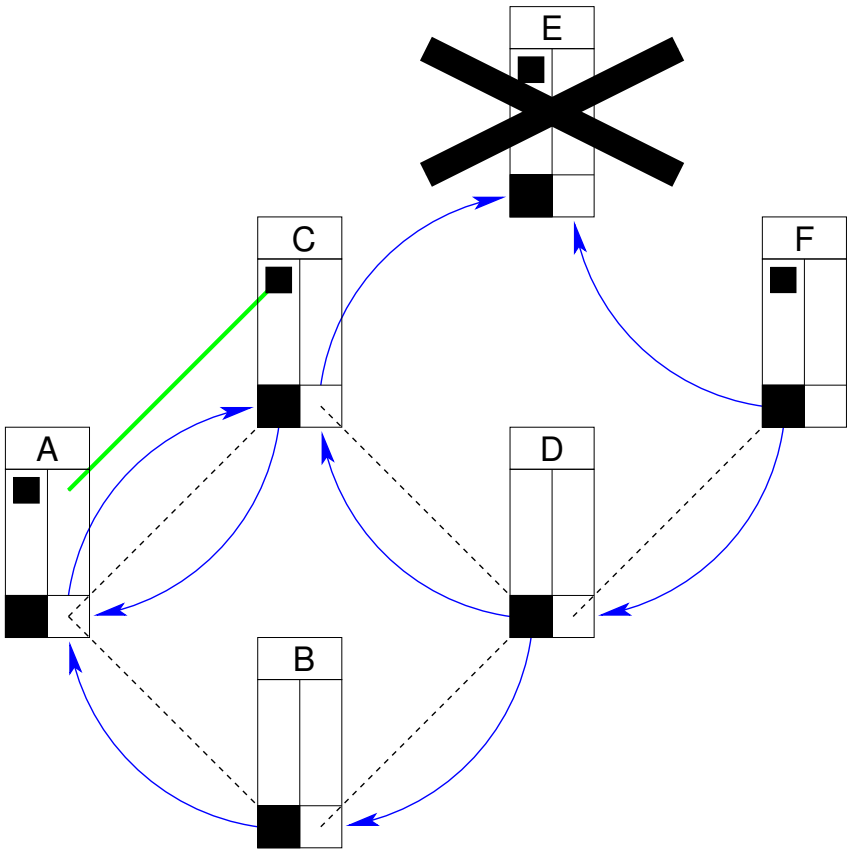
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen

Verbindungsausfall



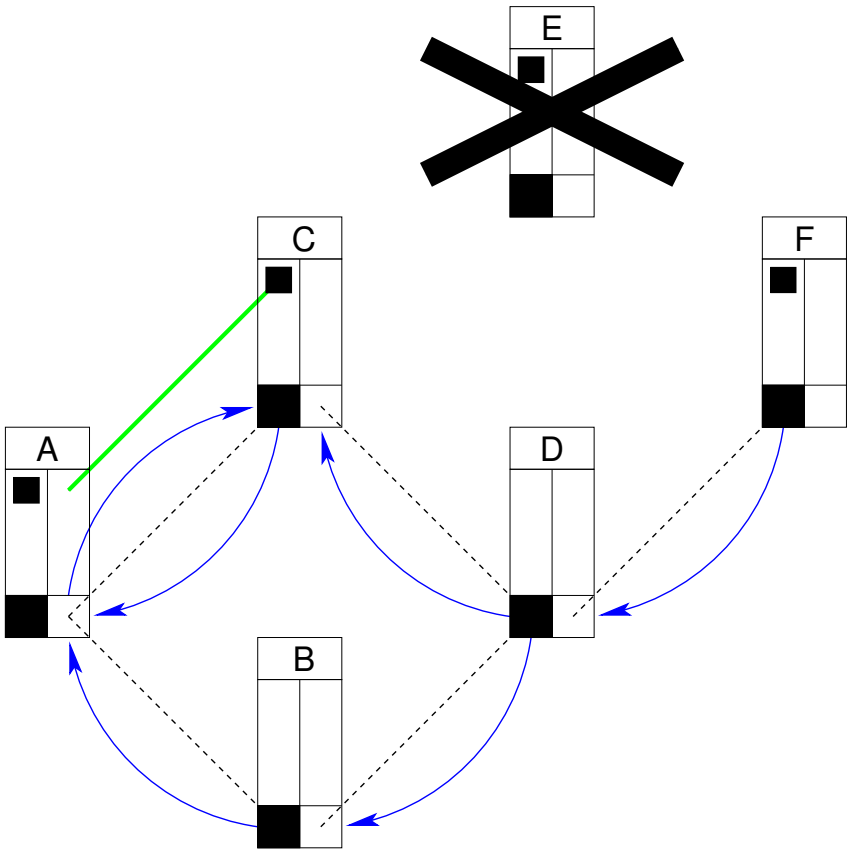
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen

Verbindungsausfall



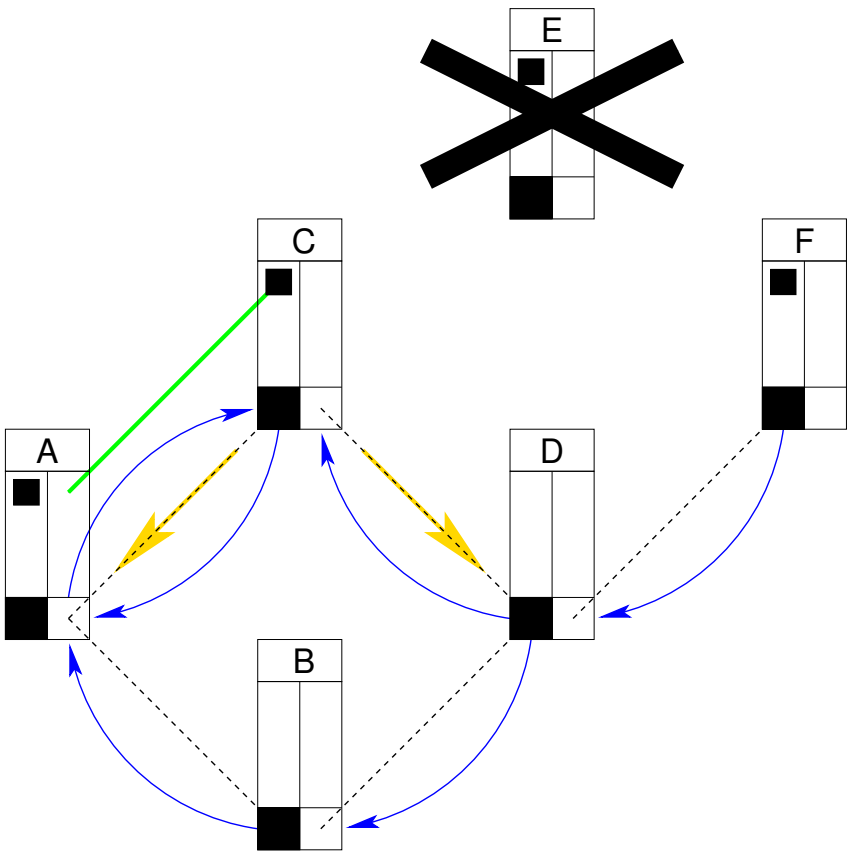
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren

Verbindungsausfall



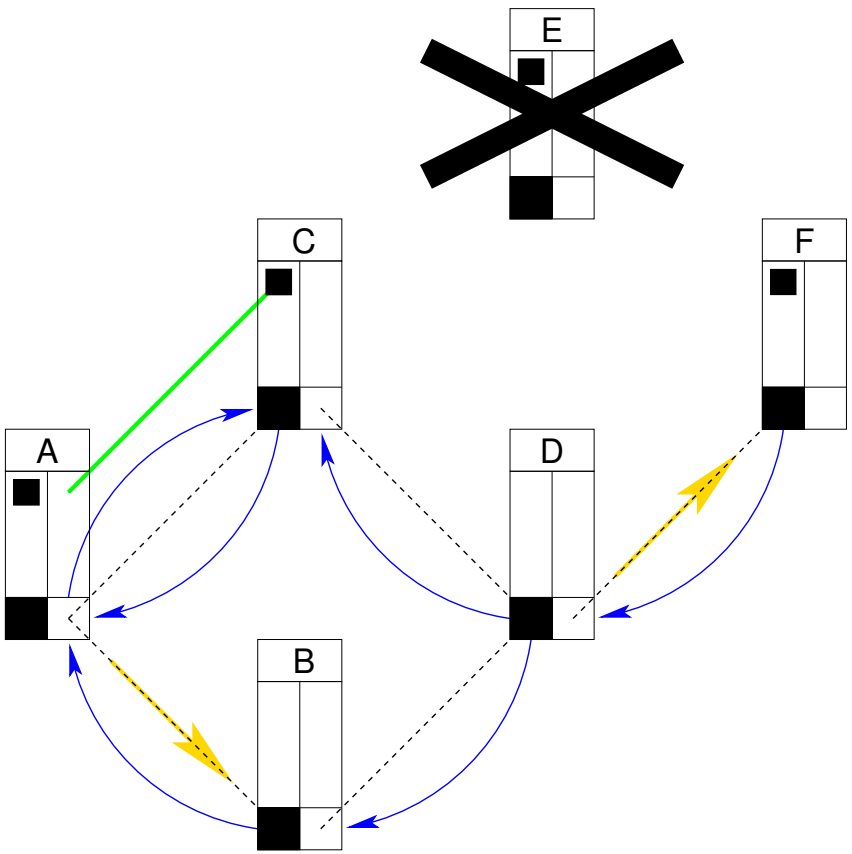
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren

Verbindungsausfall



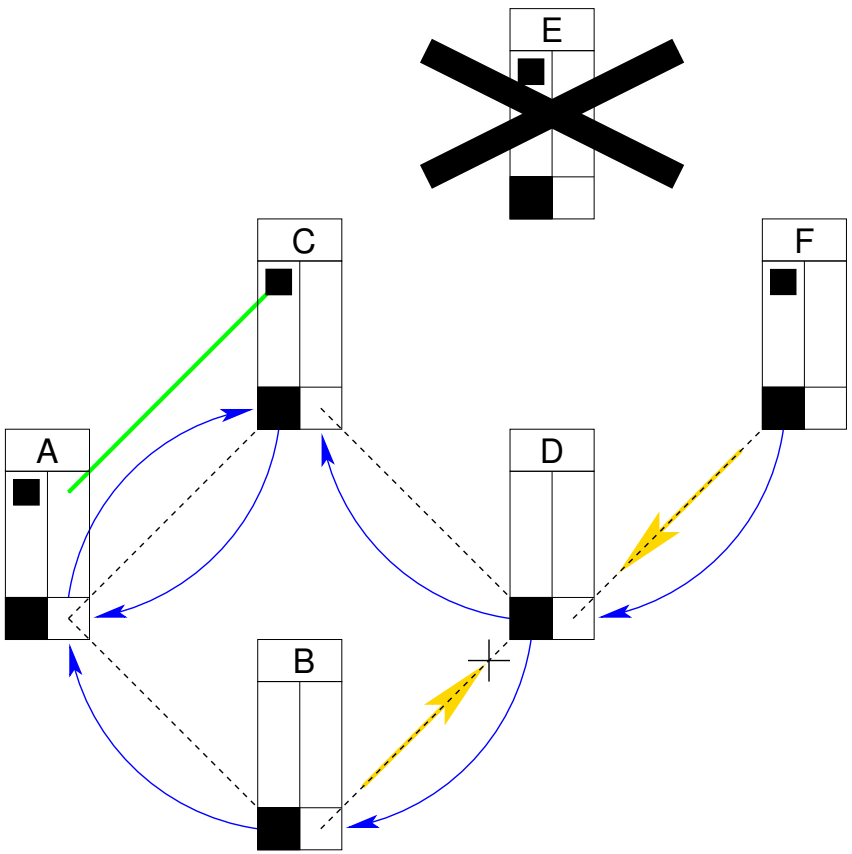
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren
- C flutet neue Anfrage

Verbindungsausfall



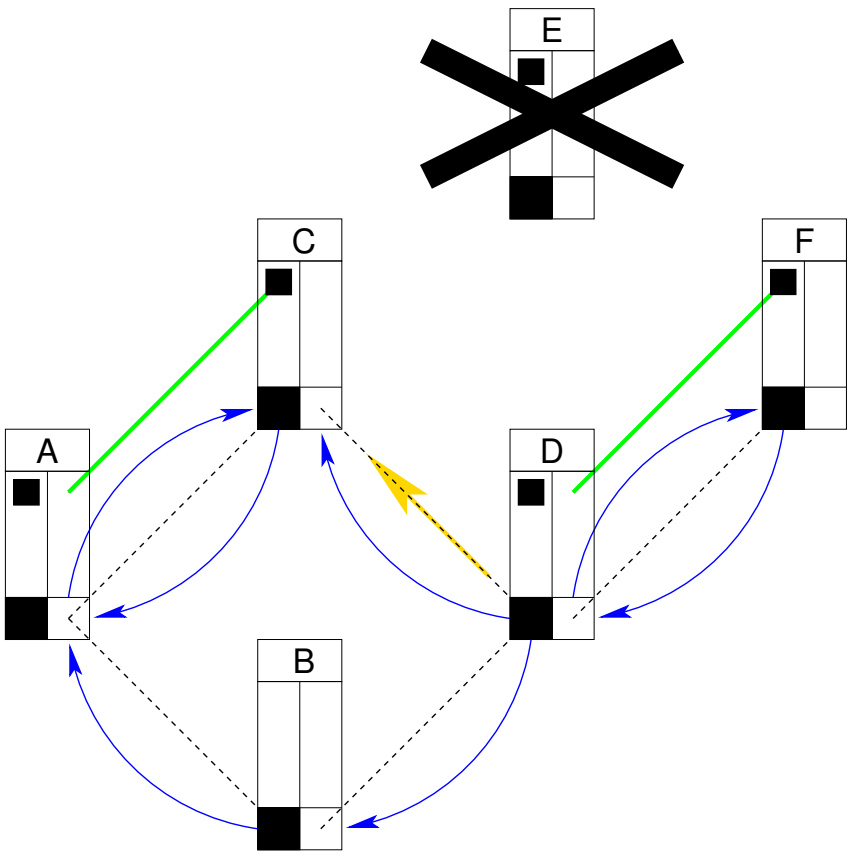
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren
- C flutet neue Anfrage
- F empfängt Anfrage

Verbindungsausfall



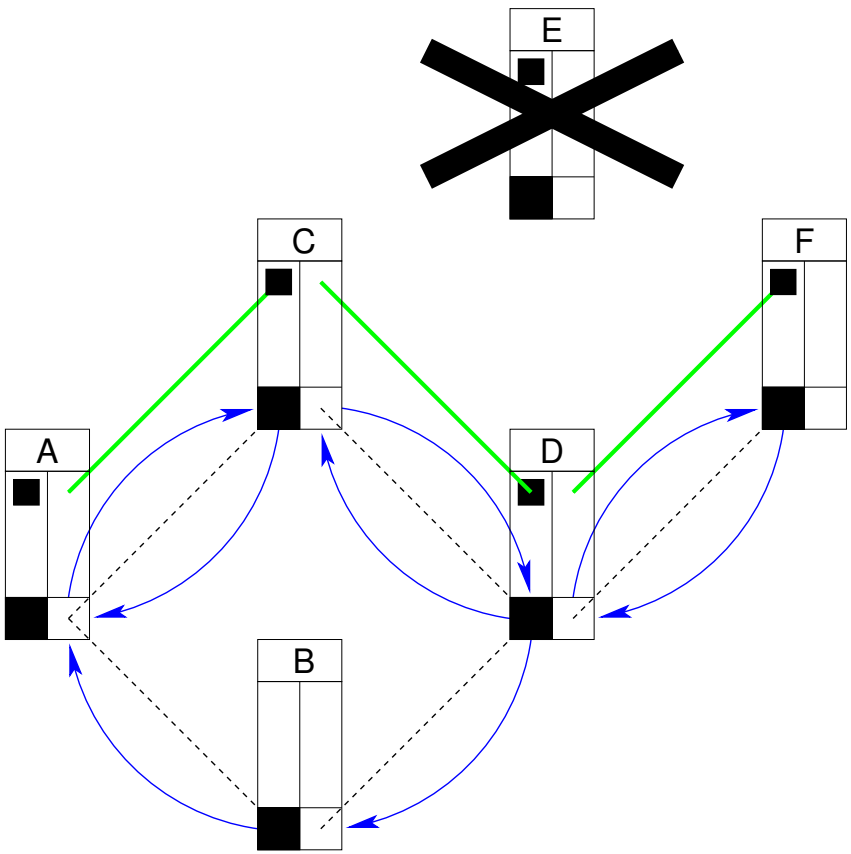
- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren
- C flutet neue Anfrage
- F empfängt Anfrage
- F sendet Antwort

Verbindungsausfall



- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren
- C flutet neue Anfrage
- F empfängt Anfrage
- F sendet Antwort
- Teilverbindung wird aufgebaut
- D leitet Antwort weiter

Verbindungsausfall



- Knoten E fällt aus
- C u. F detektieren den Ausfall
  - Datenverbindung abbauen
  - Signalisierungsverbindung abbauen
  - Routingtabelleneinträge deaktivieren
- C flutet neue Anfrage
- F empfängt Anfrage
- F sendet Antwort
- Teilverbindung wird aufgebaut
- D leitet Antwort weiter
- C empfängt Reply
- Die Verbindung ist repariert

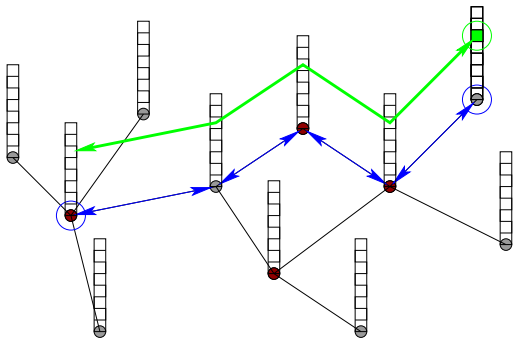
Praktischer Nutzen

- Bluetooth Kommunikationsreichweite effektiv vergrößert
- Kommunikation mit mehr als 8 Geräten möglich
- Für Nutzer Transparente Lösung
- Kompensiert Dynamik im Netz
- Kein Kommunikationsoverhead bei bestehender Verbindung (Switching)

Zusammenfassung

Verbindungsorientierter Ansatz

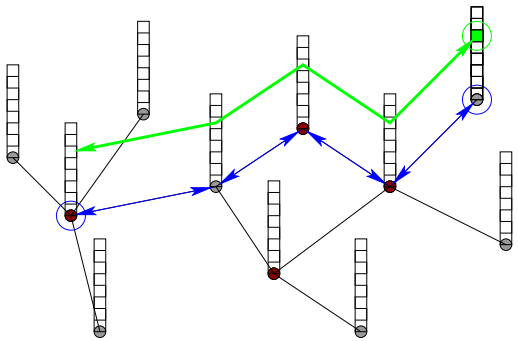
Kein Routingoverhead nach Verbindungsaufbau (Switching)



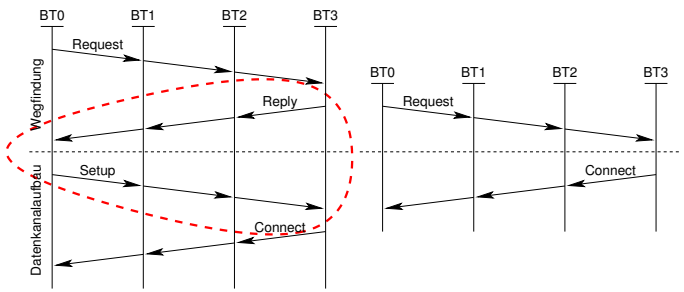
Zusammenfassung

Verbindungsorientierter Ansatz

Kein Routingoverhead nach Verbindungsaufbau (Switching)



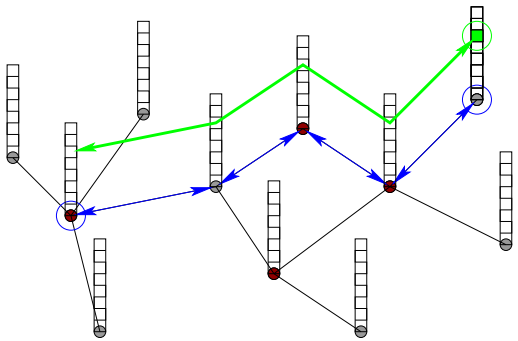
Hinweg	Topologieausprägung
	Wegfindung
	Dienstsuche
Rückweg	Verbindungsaufbau



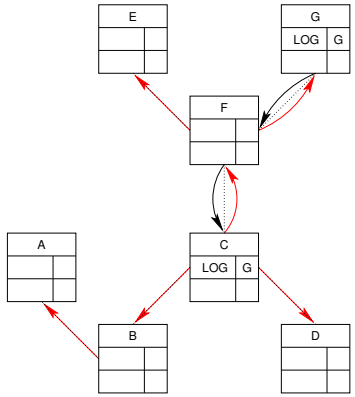
Zusammenfassung

Verbindungsorientierter Ansatz

Kein Routingoverhead nach Verbindungsaufbau (Switching)

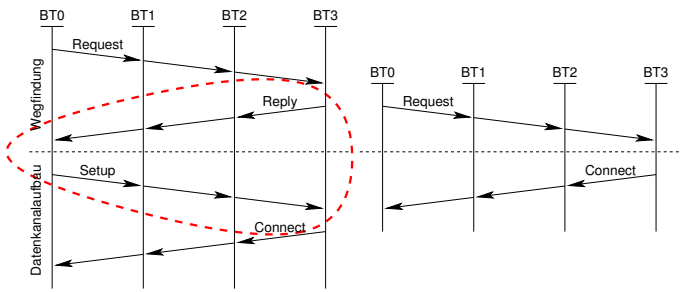


Multihop Dienstsuche



Hinweg — Topologieausprägung
Wegfindung
Dienstsuche

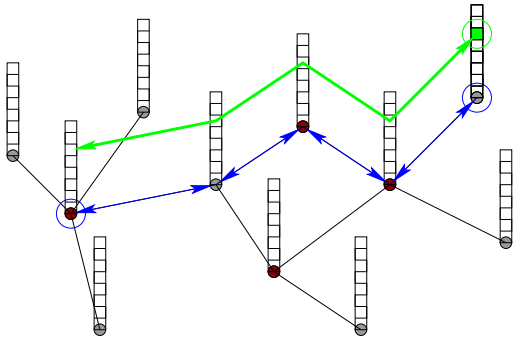
Rückweg — Verbindungsaufbau



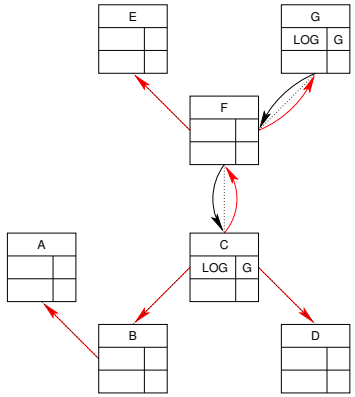
Zusammenfassung

Verbindungsorientierter Ansatz

Kein Routingoverhead nach Verbindungsaufbau (Switching)

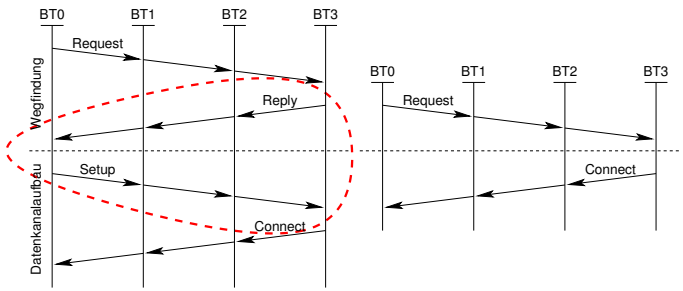


Multihop Dienstsuche

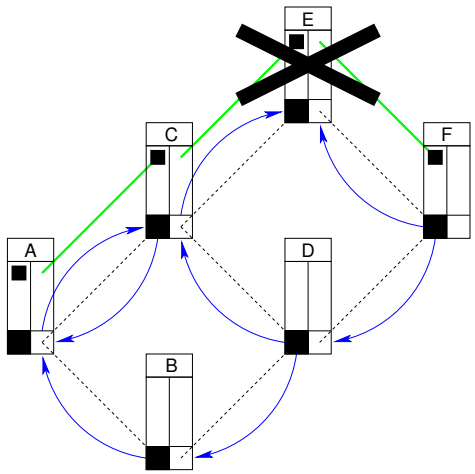


Hinweg — Topologieausprägung, Wegfindung, Dienstsuche

Rückweg — Verbindungsaufbau



Fehlerbehandlung



Ausblick

- Authentifikation
- Alternativpfad auf Standby
- Roaming
  - geordnete Verbindungsübergabe
- Quality of Service
  - Zusicherung von Übertragungsrate
  - Priorisierung der Datenströme
- RFCOMM Datenkanäle

Fragen

Anhang

- Parameter
- SNR Design Überblick
- SNR Features
- Evaluationsergebnisse
- Verwandte Arbeiten

Parameter

doRandomBroadcast: Randomisiere Broadcast

preferActiveConnections: Bevorzuge bestehende Verbindungen für Broadcast-Nachrichten

intelligentBroadcast: Unicast falls Empfänger 1 Hop entfernt. Nimmt Sender des Pakets vom Broadcast aus.

alternateMasterSlave: Sorge für alternierende Master/Slave-Verteilung bei Aufbau einer neuen Route

doLocalRepair: Lokaler Reparaturversuch bei Verbindungsverlust

acknowledgeSigMsg: Erzwingt ACKs für alle Signalisierungsnachrichten

Timeouts: Für offene Sig.Verbindungen und Routingtabelleneinträge

SNR Design Überblick

Kommunikation	Verbindungsorientiert
Signalisierung	Out of Band Signalisierung
Topologieausprägung	On Demand
Wegfindung	AODV (hier: bevorzugt schnellen Verbindungsaufbau) Verteilte Pfadinformationen
Datenkanalaufbau	gemeinsam mit Routing
Dienstsuche	Dienstbezeichner + Hostadresse Dienstbezeichner, Anycast
Fehlerbehandlung	AODV-ähnlich (siehe Szenario)

SNR Features

- Multihopkommunikation für Bluetooth
- Automatische Reparatur unterbrochener Pfade
- Ad hoc Fähigkeit
- Platformunabhängigkeit
- Funktionsfähigkeit in heterogenem Umfeld
- Keine Veränderungen gegenüber der Bluetooth Spezifikation
- Funktionsfähige Implementierung
- Verbindungsaufbau und -verwaltung
- Kein Routing Overhead nach Verbindungsaufbau
- Multihop Dienstsuche
- Über schlanke API ansprechbar
- Parametrisierbar

Evaluationsergebnisse

Round Trip Time

	Ø [ms]	σ	pro Hop
1 Hop	102 ± 0	12	51
2 Hop	261 ± 2	53	131
3 Hop	430 ± 2	72	143
4 Hop	600 ± 3	95	150
5 Hop	719 ± 3	109	144
6 Hop	858 ± 4	114	143
7 Hop	1046 ± 4	114	149

Latenz bei Verbindungsaufbau 2-10

	Ø [ms]	σ	pro Hop
1 Hop	343 ± 26	78	172
2 Hop	831 ± 41	122	416
3 Hop	1330 ± 36	109	443
4 Hop	1791 ± 32	96	448
5 Hop	2220 ± 56	168	444
6 Hop	2716 ± 94	283	453
7 Hop	3022 ± 59	117	432

Latenz bei Verbindungsaufbau 1

1 Hop	764 ms
2 Hop	3111 ms
3 Hop	4137 ms
4 Hop	6307 ms
5 Hop	7518 ms
6 Hop	9142 ms
7 Hop	16185 ms

- 10 Verbindungen zu je 100 Datenpaketen
- Ohne GUI, minimale Kontrollausgaben
- Angaben in Millisekunden
- Konfidenzintervall: $\bar{x} \pm \frac{\sigma}{\sqrt{n}}$
- Standardabweichung: $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})}$

Verwandte Arbeiten

Name	Sim/Impl	Routing	Scatternet Topologie	Platform	Besonderheiten
RVM		Source Routing	gegeben	BT Radio	Labels statt BT Adressen
ZRP	Simulation	Zone Routing	gegeben	über Link Manager	Im Nahbereich proaktiv, außerhalb reaktiv (AODV)
XHop	Implementierung	minimales Source Routing	gegeben	L2CAP BTNodes	Multihop Message Passing statt Routing, mehrere BT Interfaces
BlueTree		Binäre Suche	BlueTree (statisch)	N/A	Bin-Baum auf BT Adressen, Wurzelknoten verwaltet SN, Knoten in Funkreichweite
SRS	Simulation	ähnlich AODV	On Demand	N/A	Verändert BT Protokoll → effizienter Broadcast, → Piconetzweite Synch.
Caspar	Implementierung	Baum Suche	Baum (dynam.)	RFCOMM	Baumstruktur auf dyn. Parametern aufbaubar, z.B. Energiereserven (Future Work)
SNR	Implementierung	AODV	On Demand	L2CAP JSR-82	Multihop Service Search, Verb.-Aufbau, Verb.-Verwaltung, unveränderter BT-Stack, platformunabhängig

Verwandte Arbeiten

RVM A Routing Vector Method (RVM) for Routing in Bluetooth Scatternets, P. Bhagwat & A. Segall 1999

- Source Routing
- Labels statt Bluetooth Adressen
- Setzt auf Bluetooth Radio auf

Verwandte Arbeiten

ZRP A Zone Routing Protocol for Bluetooth Scatternets, R. Kapoor, M. Gerla 2003

- Zone Routing Protokoll. Proaktiv im Nahbereich, außerhalb reaktiv
- Setzt direkt über Link Manager auf
- Simulation

Verwandte Arbeiten

XHop Bluetooth Smart Nodes for Mobile Ad-hoc Networks, J. Beutel, O. Kasten, et al. 2003, ETH Zürich

- Minimales Dynamic Source Routing Protokoll
- Multihop Message Passing statt Routing.
- Setzt auf L2CAP auf
  - Öffnet Verbindung
  - Überträgt Paket
  - Schließt Verbindung
  - Wertet Paket aus
- Mehrere Bluetooth Interfaces
- Implementiert auf BTNodes

Verwandte Arbeiten

Blue Tree A Self-Routing Topology for Bluetooth Scatternets, M. Sun, C. Chang, T. Lai 2002

- Scatternetz zu BlueTree ausgeprägt
- Binärer Suchbaum über Bluetooth Adressen
- Wurzelknoten des Baums verwaltet das Netz
- Problem: Geräte müssen sich gegenseitig hören können $\Rightarrow$ Partitionierung

Verwandte Arbeiten

SRS A Bluetooth Scatternet-Route Structure for Multihop Ad Hoc Networks, Y. Liu, J. Lee, T. Saadawi, 2003

- AODV- ähnliches Routing
- Ausprägung der Topologie: on demand
- Verändert Bluetooth Spezifikation für
  - effizienten Broadcast
  - Synchronisation zwischen Piconetzen
- Simulation

Verwandte Arbeiten

Caspar M. Caspar, 2006 (noch nicht veröffentlicht)

- Funktionsfähiger Prototyp
- Baumartige Netzwerktopologie
- Geräte werden nach 'Güte' in Baum einsortiert (derzeit noch statisch)
- Routing auf Baumstruktur
- Setzt auf RFCOMM auf
- Bisher noch keine Evaluationsergebnisse verfügbar