

Planning of Time Triggered Communication Schedules

Kommunikationsplanung für echtzeitfähige Kommunikationsnetzwerke
19.11.2009

Olaf Graeser, Oliver Niggemann

inIT - Institut Industrial IT
Hochschule Ostwestfalen-Lippe
32657 Lemgo
olaf.graeser@hs-owl.de

Agenda

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Motivation
 - Anwendungsfall
- Netzwerkgraph
- Qualitative Bewertung (eines Schedules)
- Planung
- Ergebnisse

Echtzeitkommunikation in Feldbussystemen

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Profinet IRT
 - Einsatzgebiete:
 - Prozess- und Fertigungsautomatisierung
 - Motion Control (Antriebstechnik)

Prozessautomatisierung

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Kommunikationszyklus > 100 ms
- geringe Echtzeitanforderungen



Quelle: <http://www.processlogics.net/page2>

Fertigungsautomatisierung

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Kommunikationszyklus ca. 10 ms
- mittlere Echtzeitanforderungen

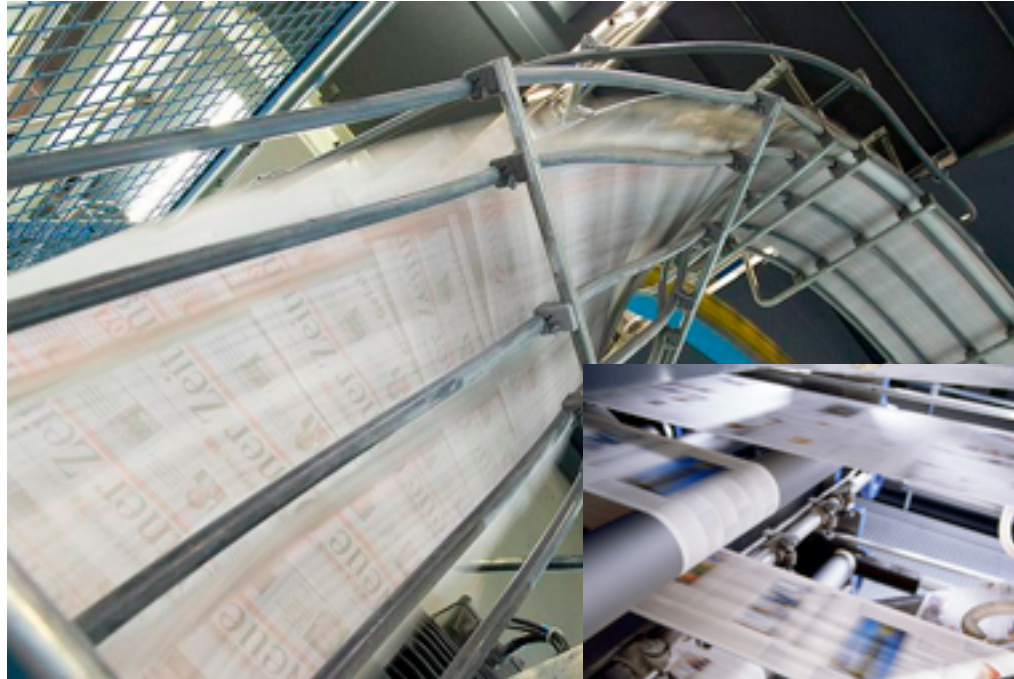


Quelle: <http://sigmaco.ir>

Motion Control (Antriebstechnik)

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Kommunikationszyklus < 1 ms
- hohe Echtzeitanforderungen



Quelle: <http://www.zehnder.ch/index.php?id=235>



Quelle: <http://www.topsolid.de/successes/report/kba.asp>

Echtzeitkommunikation in Feldbussystemen

➔ *Motivation*

➔ *Netzwerkgraph*

➔ *Schedule-Qualität*

➔ *Planung*

➔ *Ergebnisse*

Echtzeitkommunikation in Feldbussystemen

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Profinet IRT
 - Einsatzgebiete:
 - Prozess- und Fertigungsautomatisierung
 - Motion Control (Antriebstechnik)

Echtzeitkommunikation in Feldbussystemen

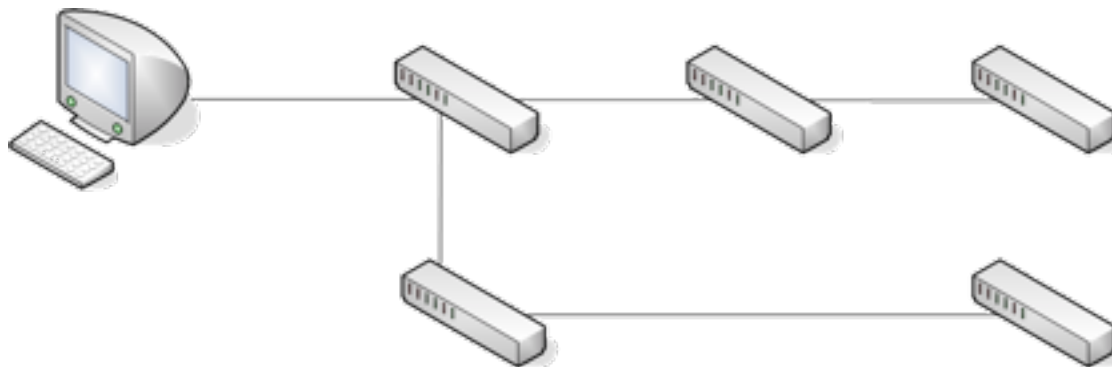
→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Profinet IRT
 - Einsatzgebiete:
 - Prozess- und Fertigungsautomatisierung
 - Motion Control (Antriebstechnik)
 - Switched Ethernet
 - Typische Topologie: Linie mit Abzweigungen
 - Mögliche Topologie: Vermaschtes Netz

Echtzeitkommunikation in Feldbussystemen

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

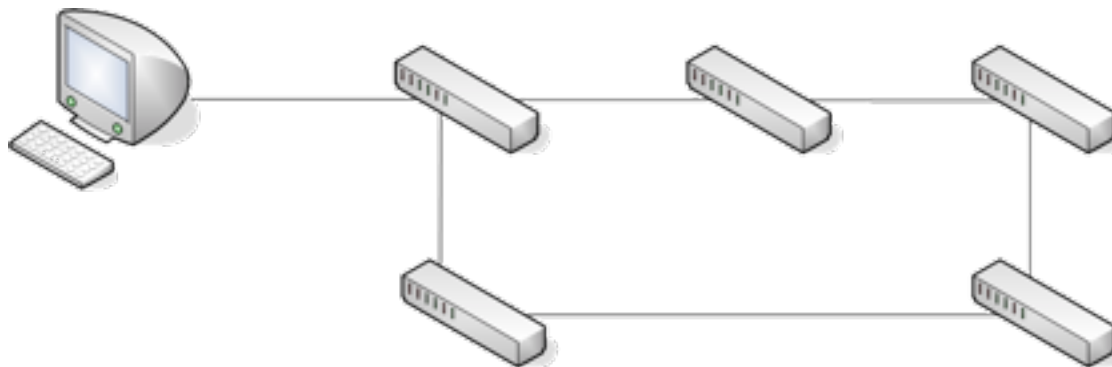
- Profinet IRT
 - Einsatzgebiete:
 - Prozess- und Fertigungsautomatisierung
 - Motion Control (Antriebstechnik)
 - Switched Ethernet
 - Typische Topologie: Linie mit Abzweigungen
 - Mögliche Topologie: Vermaschtes Netz



Echtzeitkommunikation in Feldbussystemen

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Profinet IRT
 - Einsatzgebiete:
 - Prozess- und Fertigungsautomatisierung
 - Motion Control (Antriebstechnik)
 - Switched Ethernet
 - Typische Topologie: Linie mit Abzweigungen
 - Mögliche Topologie: Vermaschtes Netz



Echtzeitkommunikation in Feldbussystemen

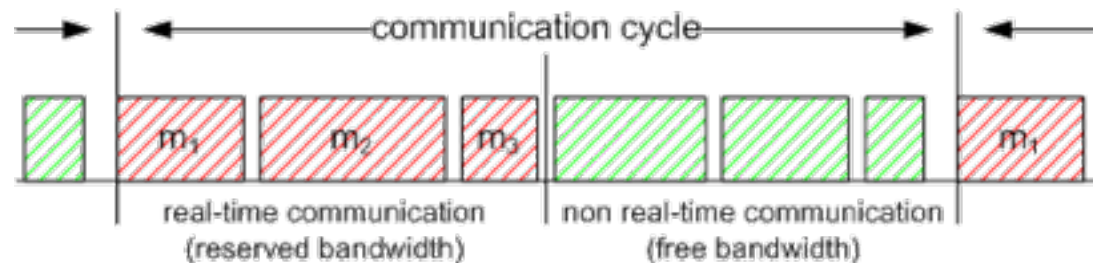
→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

- Profinet IRT
 - Kompatibel zum Standard Ethernet
 - Bandbreitenreservierung in der Sicherungsschicht (Data Link Layer)
 - Deterministisches Kommunikationssystem
 - Erfüllt harte Echtzeitanforderungen

Echtzeitkommunikation in Feldbussystemen

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

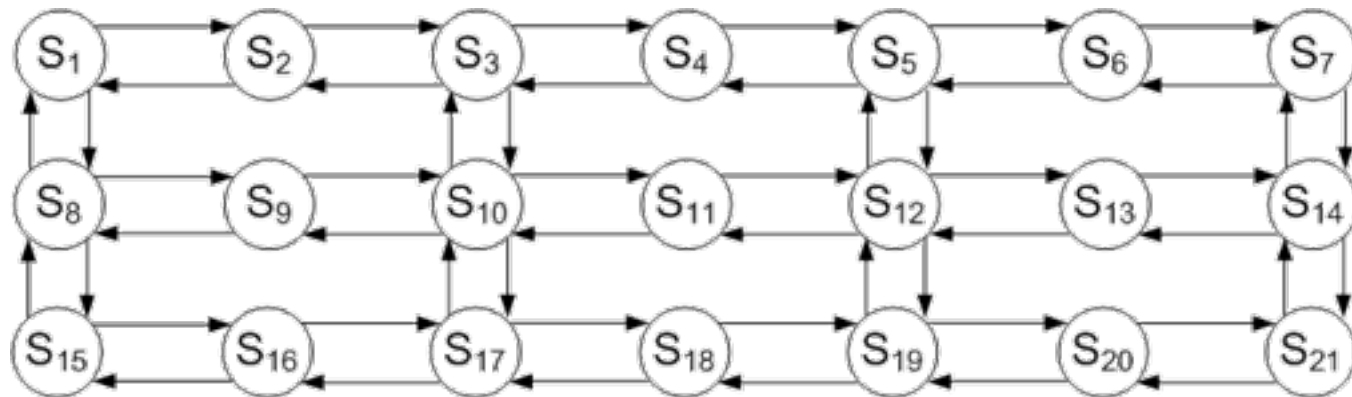
- Profinet IRT
 - Kompatibel zum Standard Ethernet
 - Bandbreitenreservierung in der Sicherungsschicht (Data Link Layer)
 - Deterministisches Kommunikationssystem
 - Erfüllt harte Echtzeitanforderungen



- 2 Phasen der Kommunikation
 - real-time (rote Phase)
 - non real-time (grüne Phase)
- Ziel: rote Phase minimieren!

Was muss geplant werden?

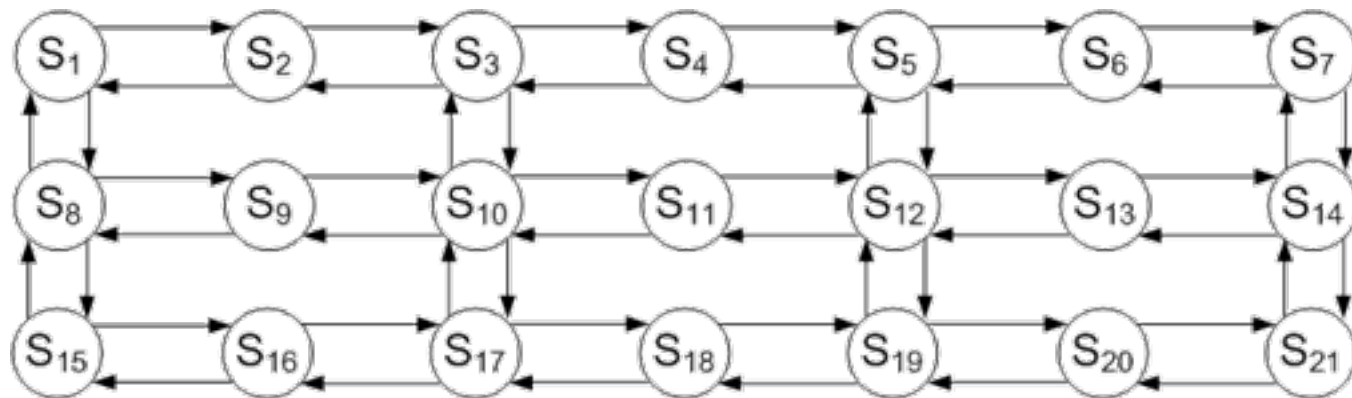
→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse



Was muss geplant werden?

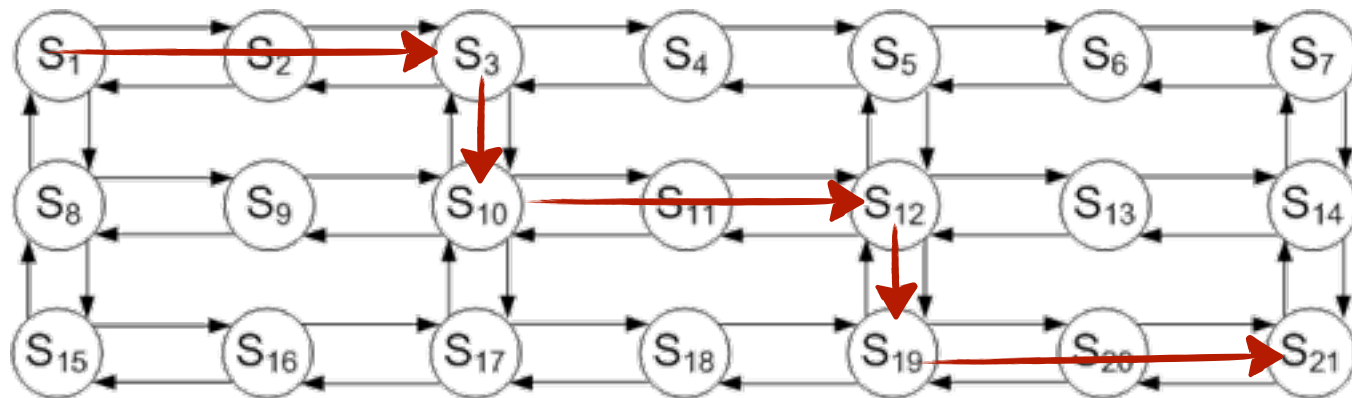
→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

- 2 Freiheitsgrade (pro Nachricht)
 - Route/Pfad



Was muss geplant werden?

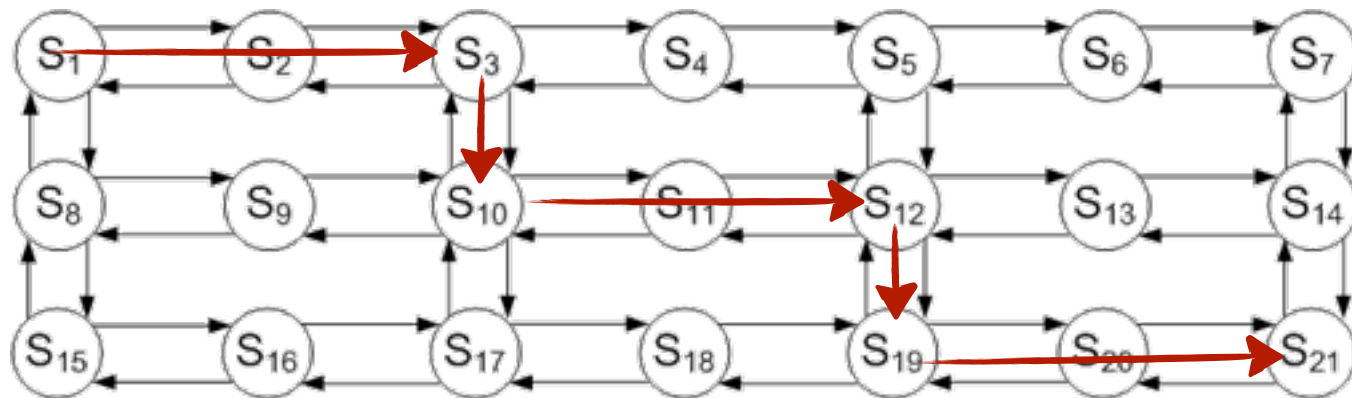
- 2 Freiheitsgrade (pro Nachricht)
 - Route/Pfad



Was muss geplant werden?

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

- 2 Freiheitsgrade (pro Nachricht)
 - Route/Pfad
 - Schedule - Reihenfolge der Nachrichten



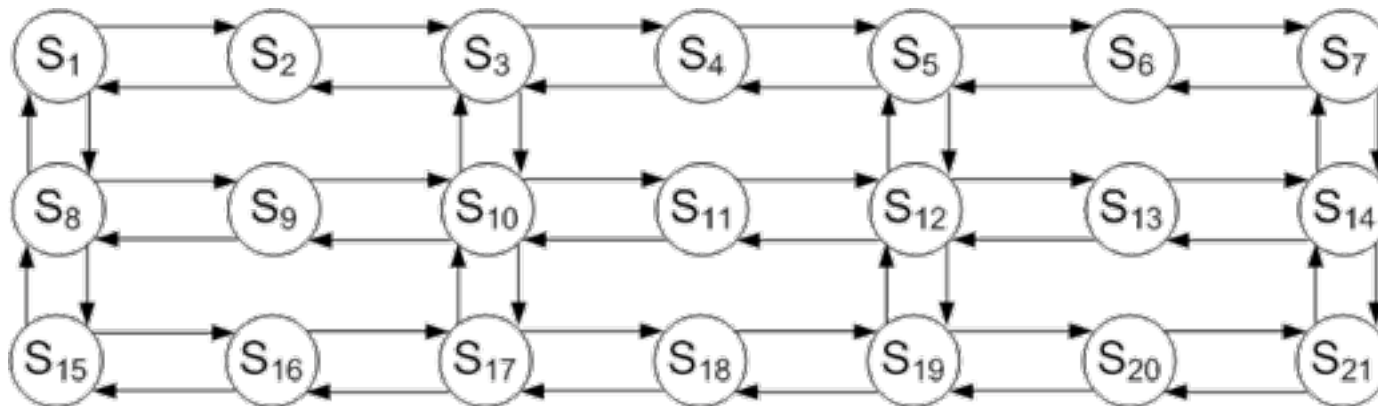
Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

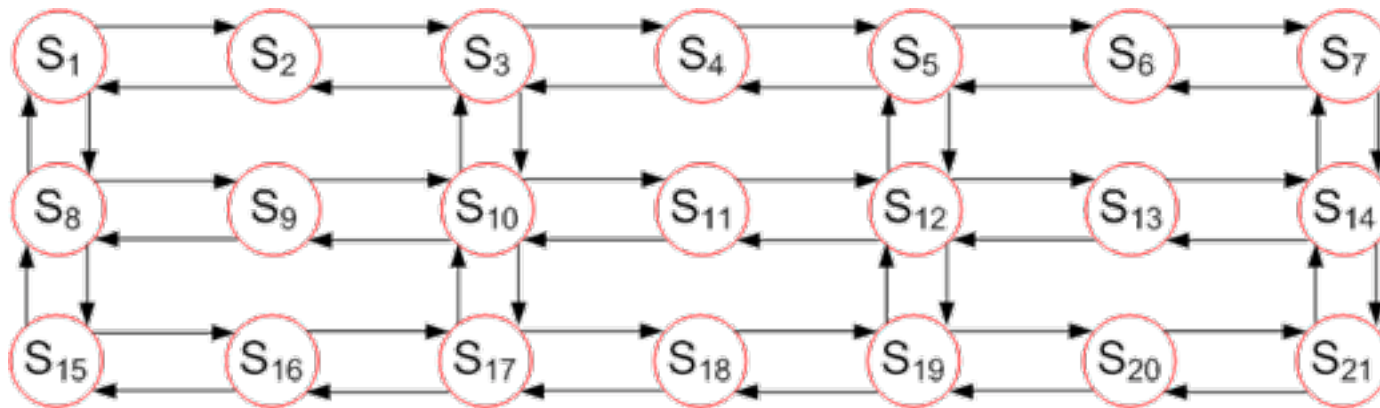
- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

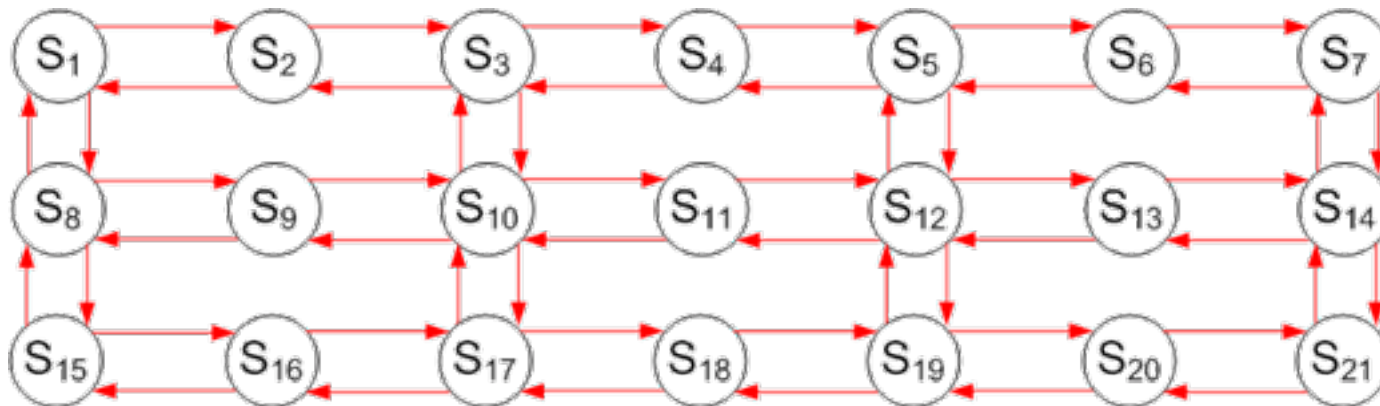
- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden



Das Netzwerk als Graph

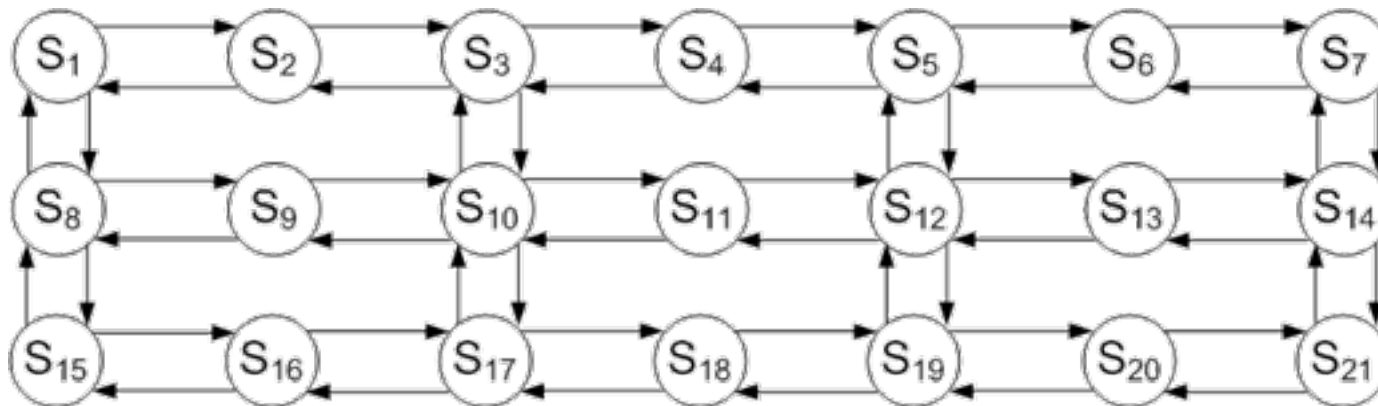
→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden



Das Netzwerk als Graph

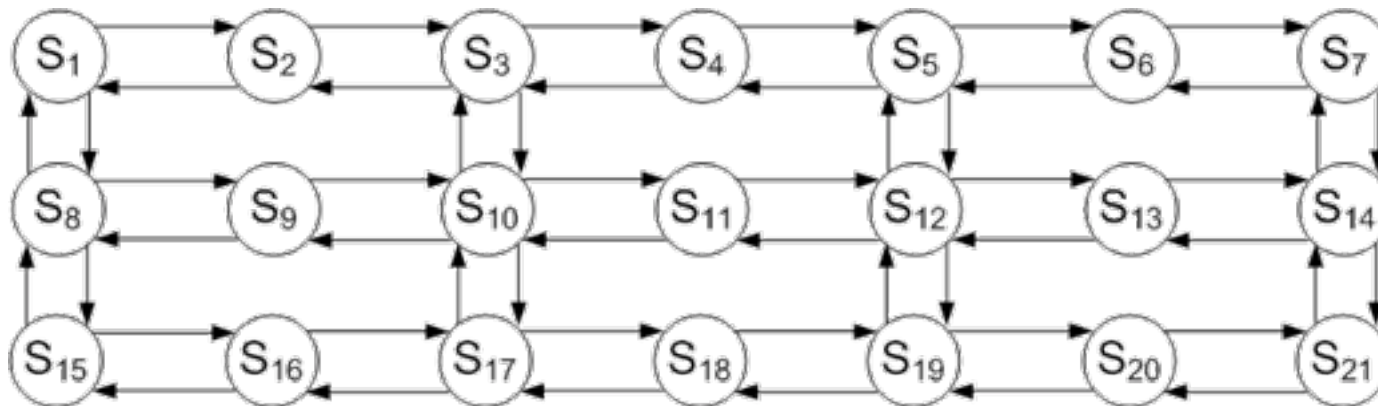
→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

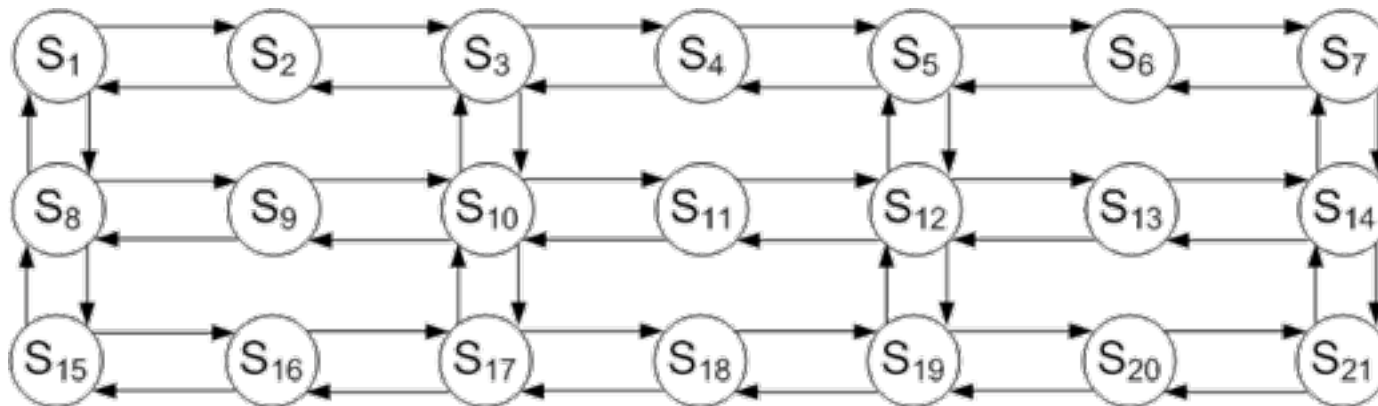
- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$



Das Netzwerk als Graph

→ Motivation

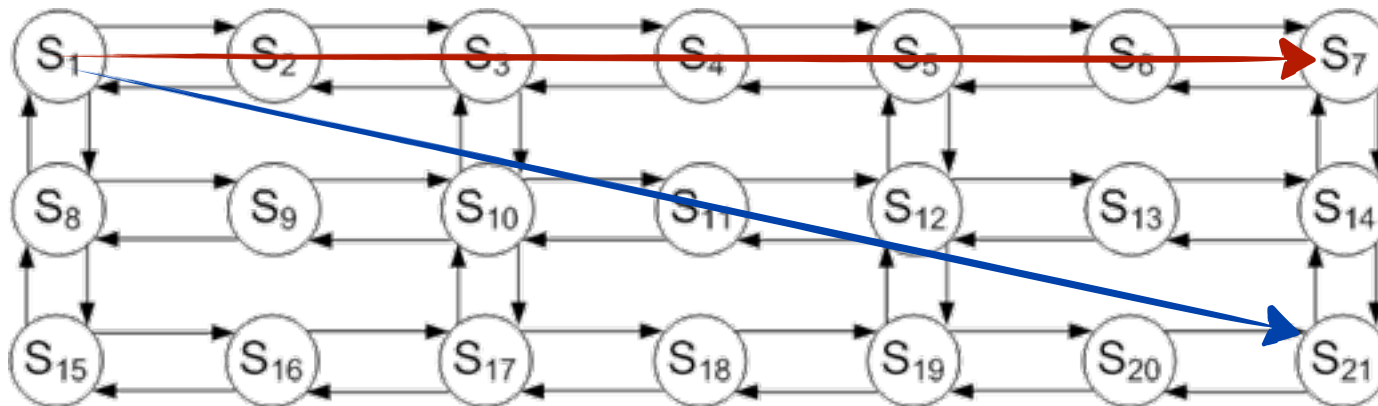
→ **Netzwerkgraph**

→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)



Das Netzwerk als Graph

→ Motivation

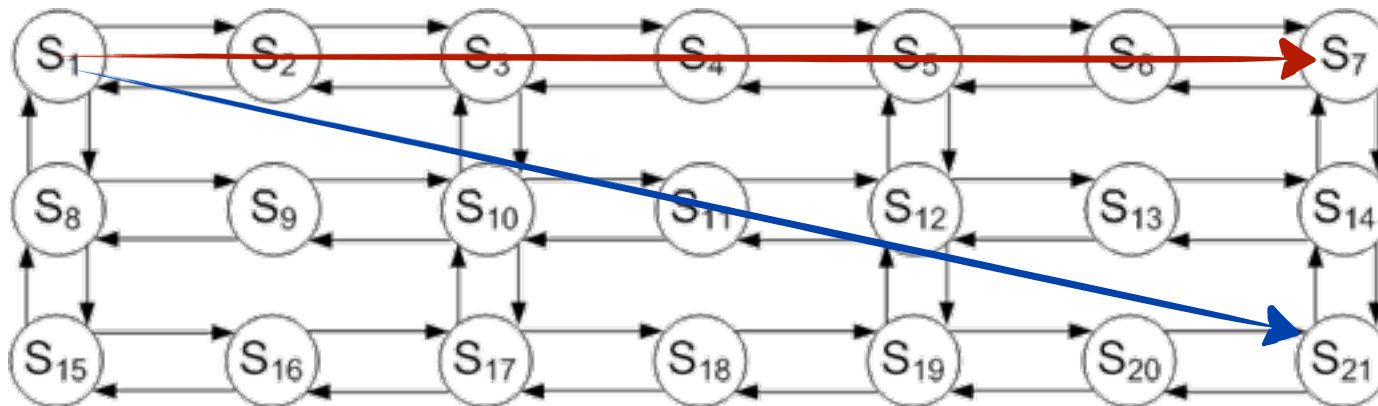
→ Netzwerkgraph

→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$



Das Netzwerk als Graph

→ Motivation

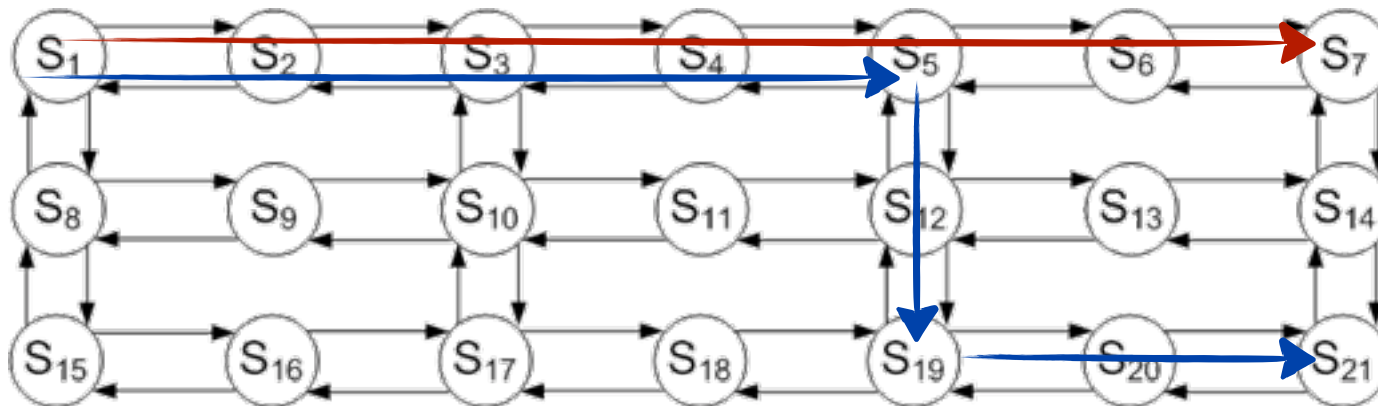
→ **Netzwerkgraph**

→ Schedule-Qualität

→ Planung

→ Ergebnisse

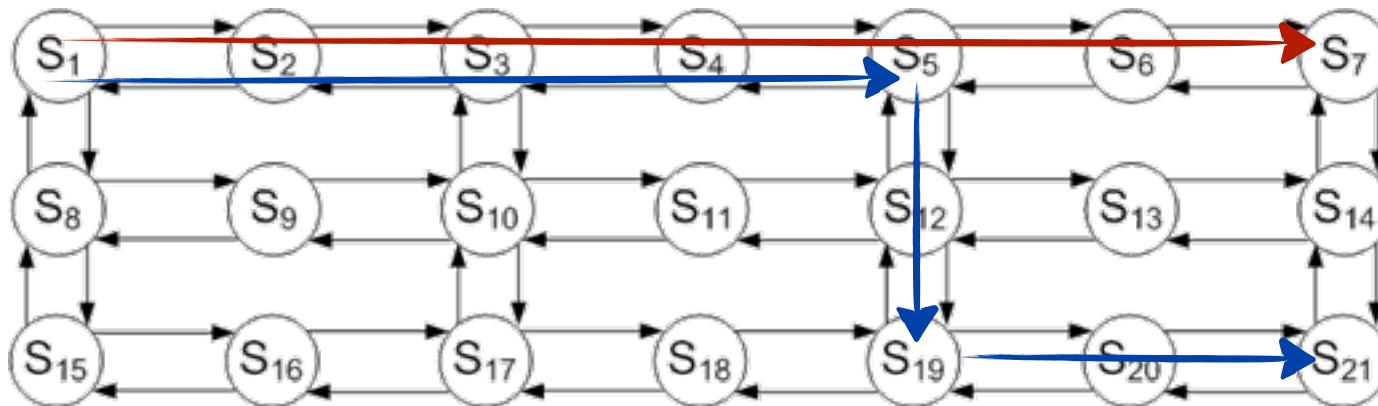
- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

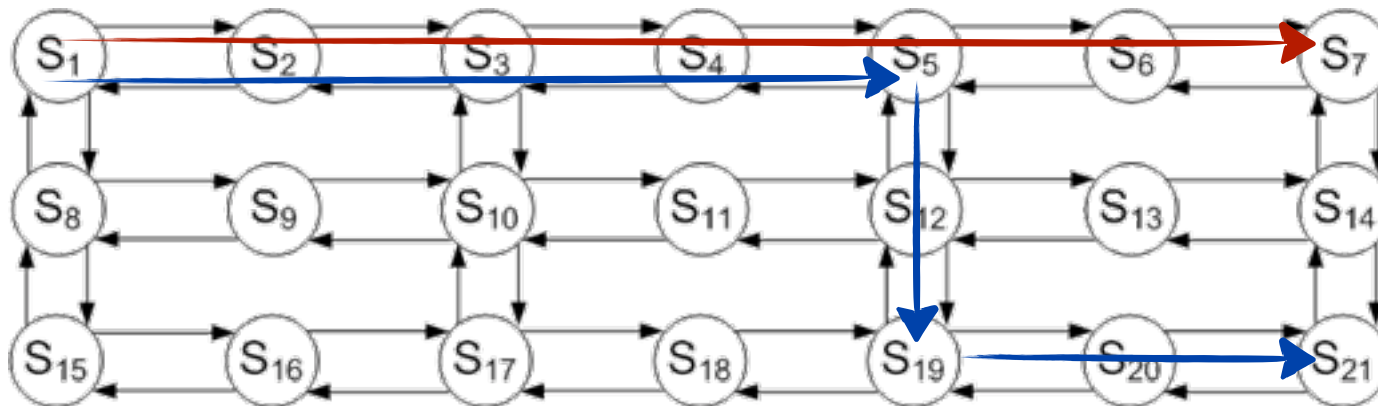
(zum Beispiel m_1 , m_2)



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$
 - Kantenbelegungsfunktion $\delta_i : p_i \rightarrow \mathbf{R}^2$



Das Netzwerk als Graph

→ Motivation

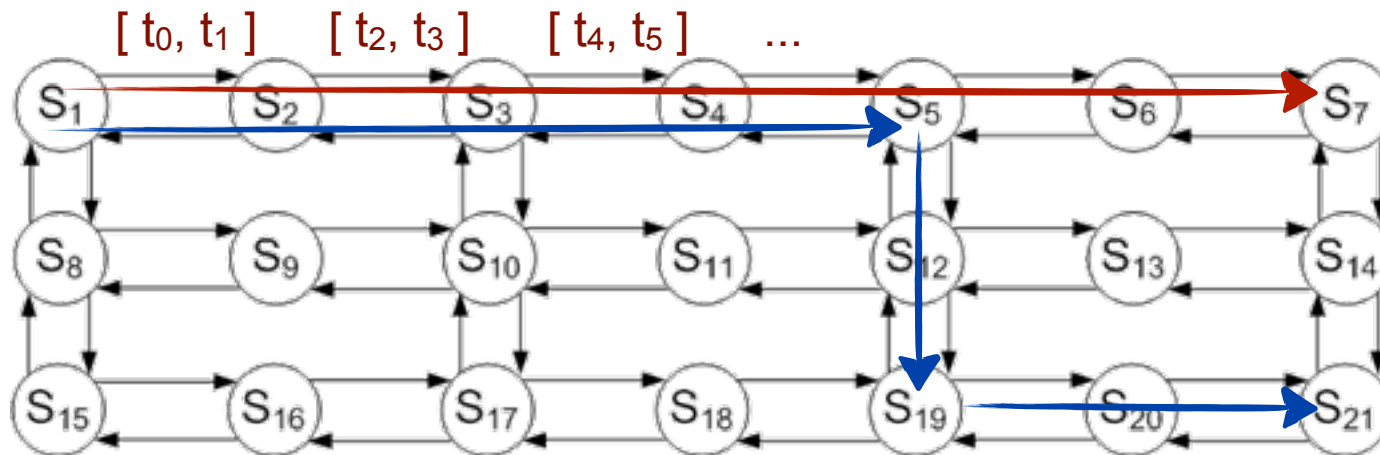
→ Netzwerkgraph

→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$
 - Kantenbelegungsfunktion $\delta_i : p_i \rightarrow \mathbf{R}^2$



Das Netzwerk als Graph

→ Motivation

→ Netzwerkgraph

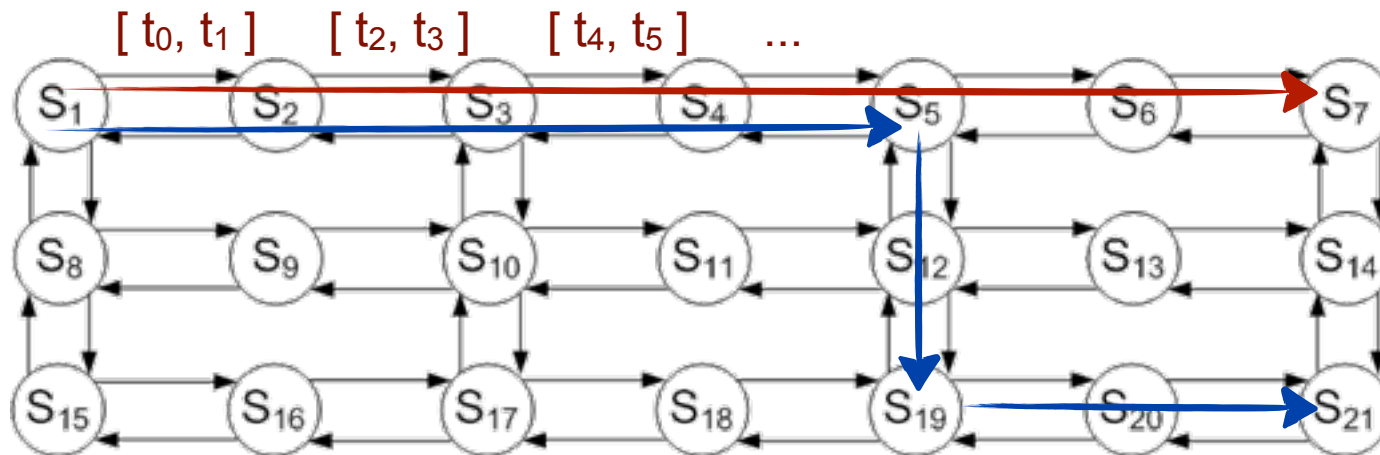
→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$
 - Kantenbelegungsfunktion $\delta_i : p_i \rightarrow \mathbf{R}^2$

$$t_2 \leq t_1$$



Cut-Through Switching

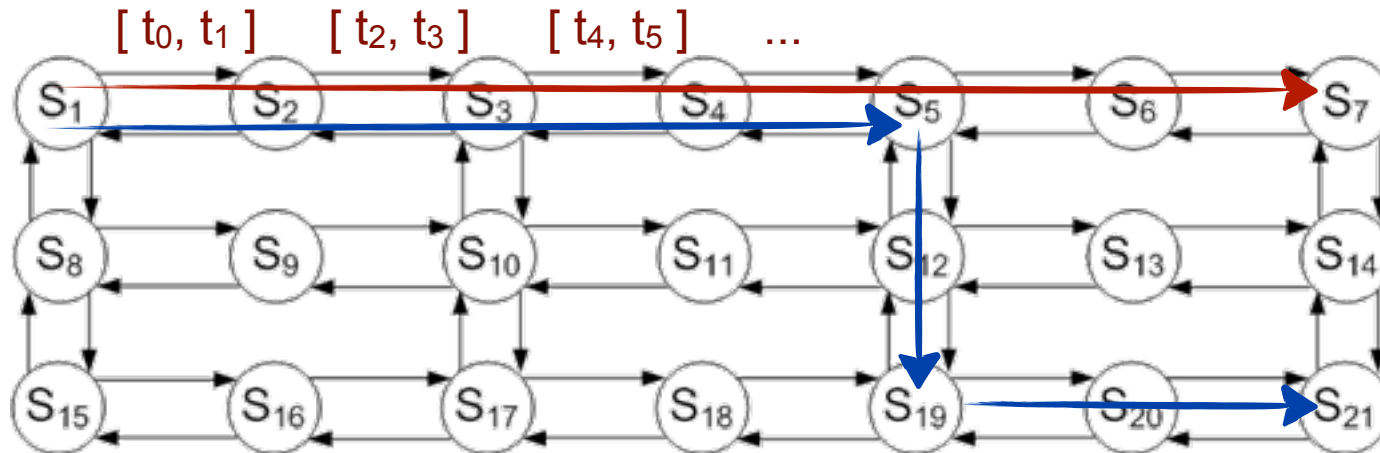
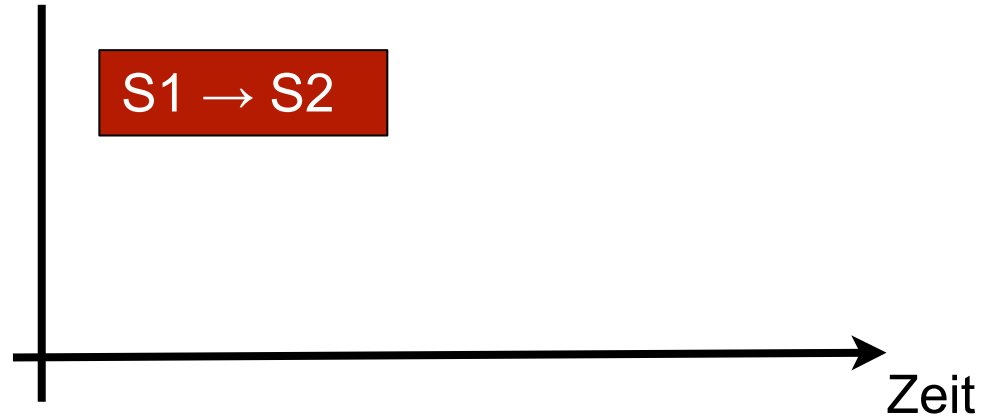
➔ Motivation

➔ **Netzwerkgraph**

➔ Schedule-Qualität

➔ Planung

➔ Ergebnisse



Cut-Through Switching

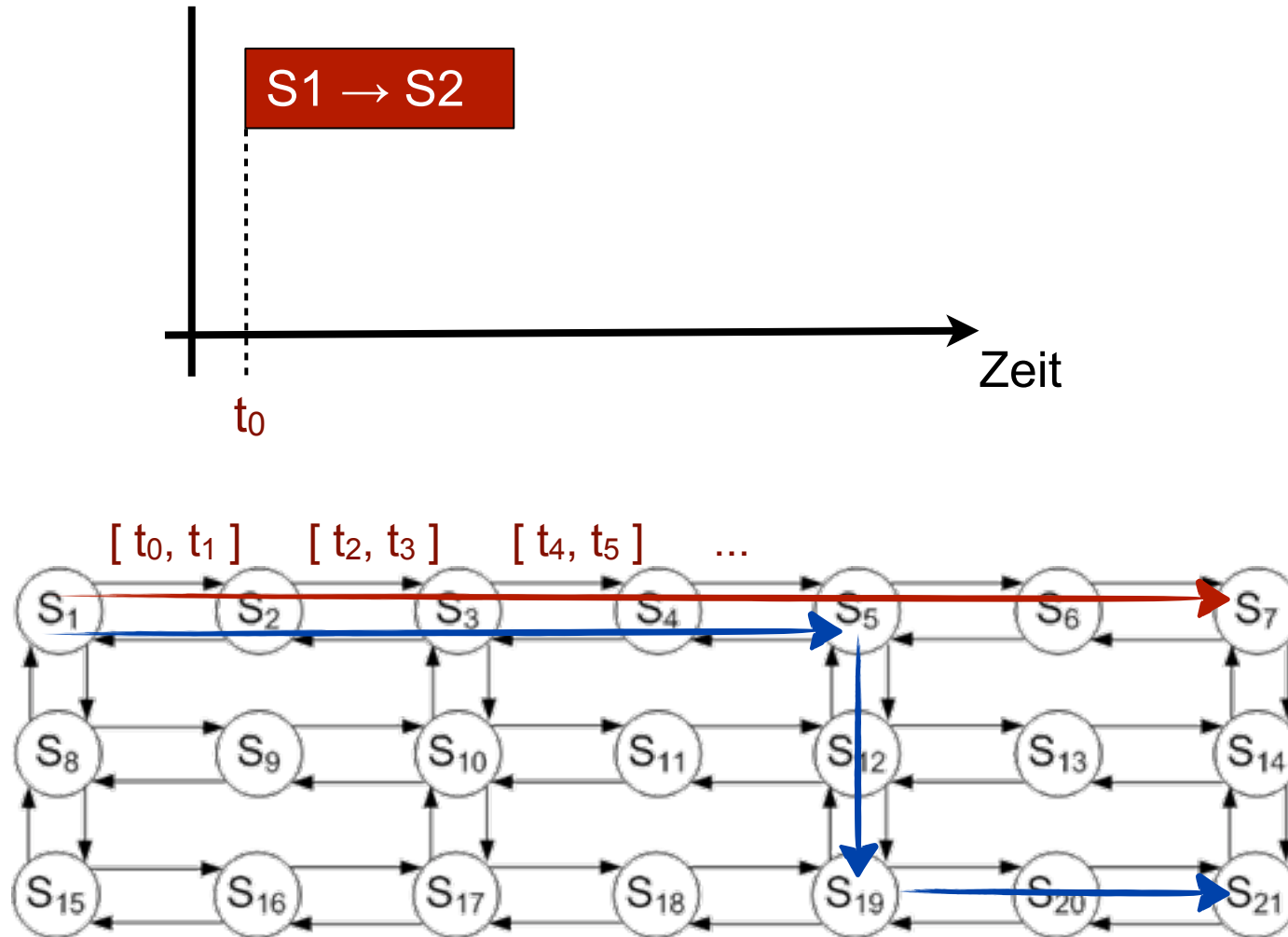
→ Motivation

→ Netzwerkgraph

→ Schedule-Qualität

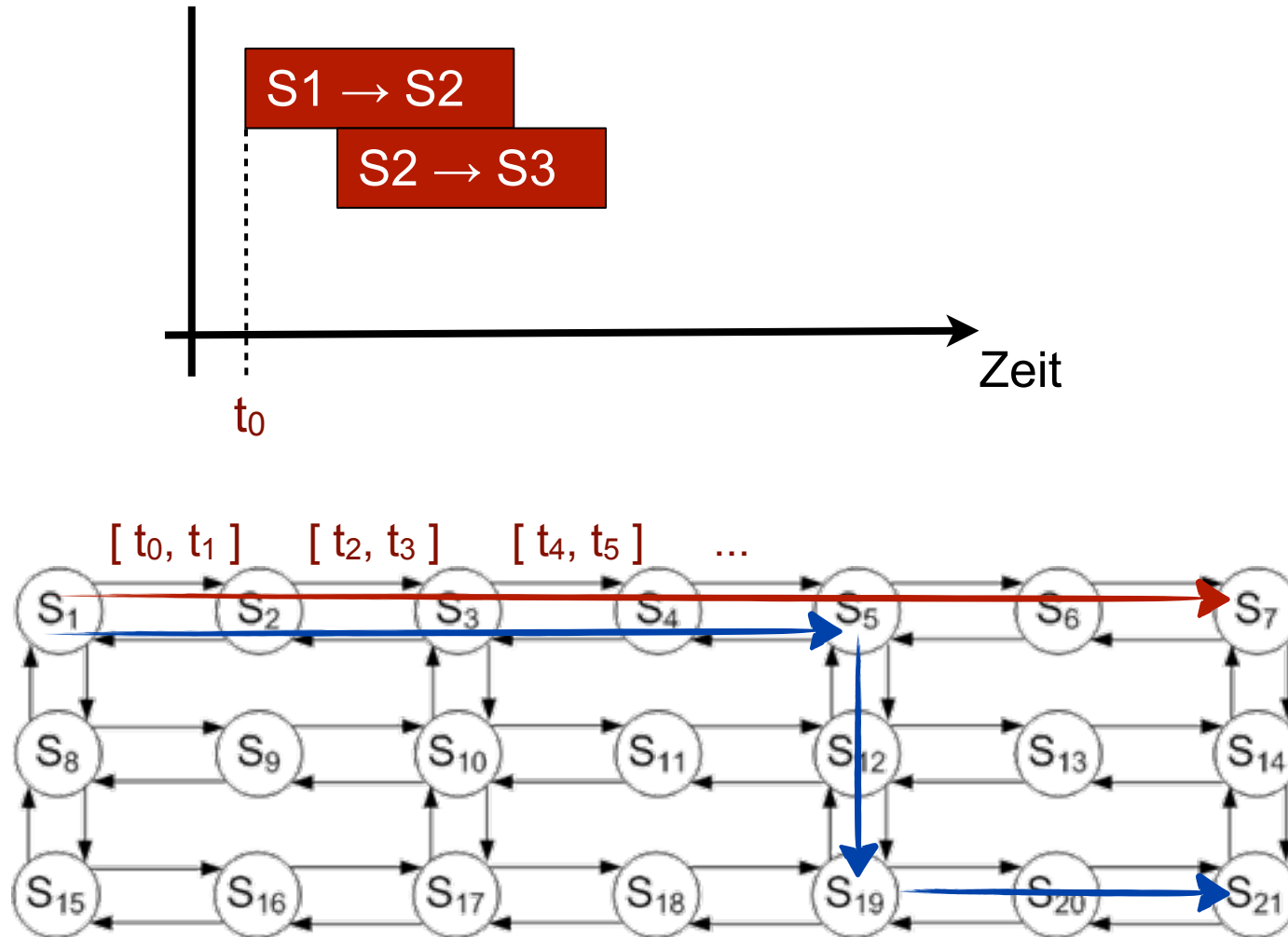
→ Planung

→ Ergebnisse



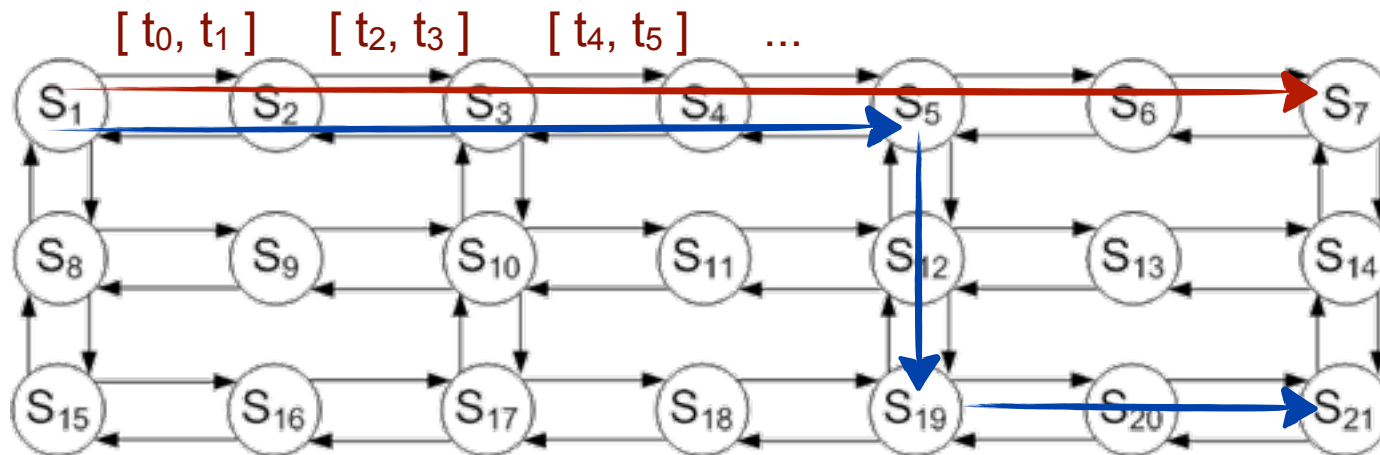
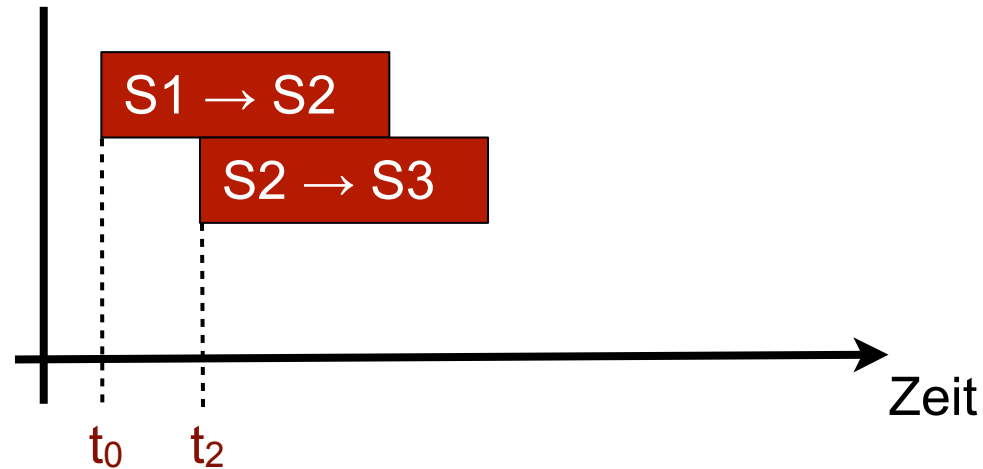
Cut-Through Switching

→ Motivation → **→ Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse



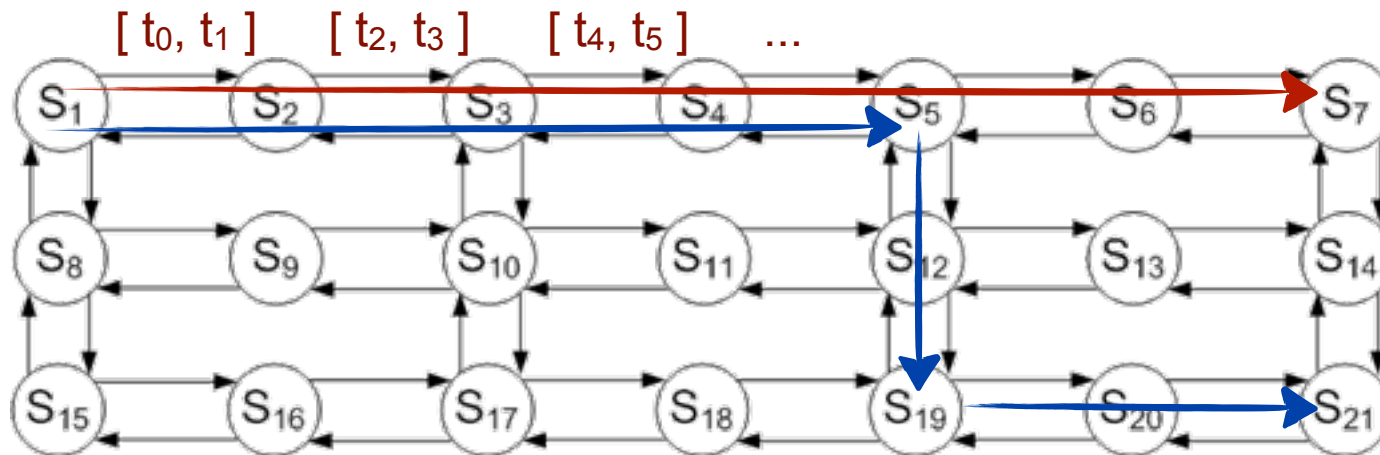
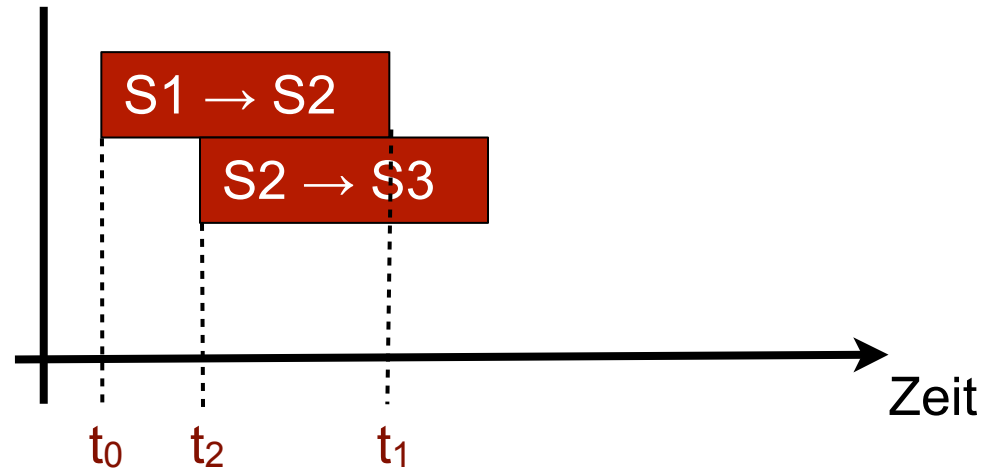
Cut-Through Switching

→ Motivation → **→ Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

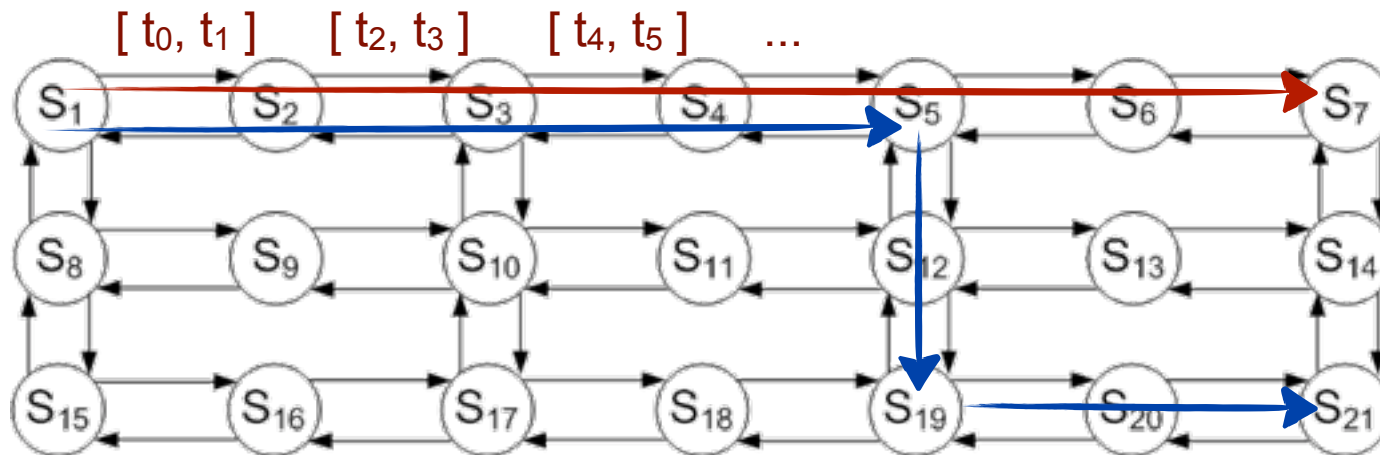
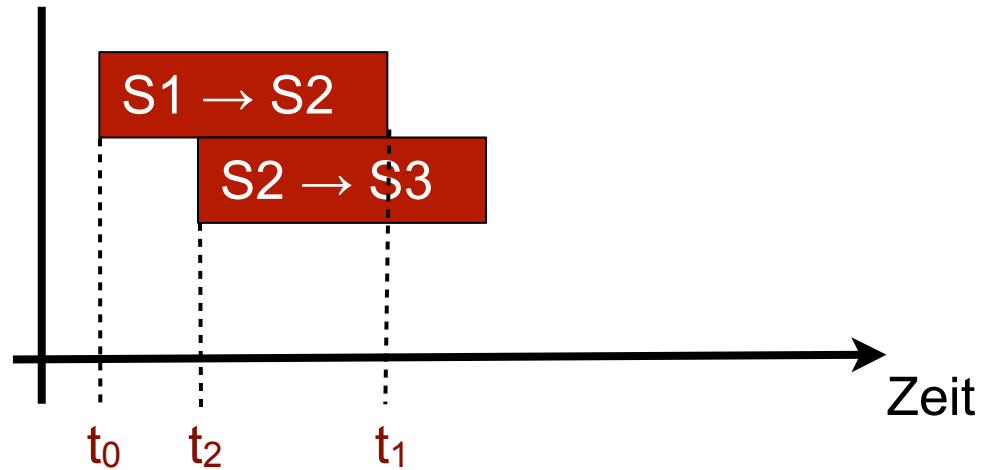


Cut-Through Switching

→ Motivation → **→ Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse



Cut-Through Switching



Cut-Through Switching

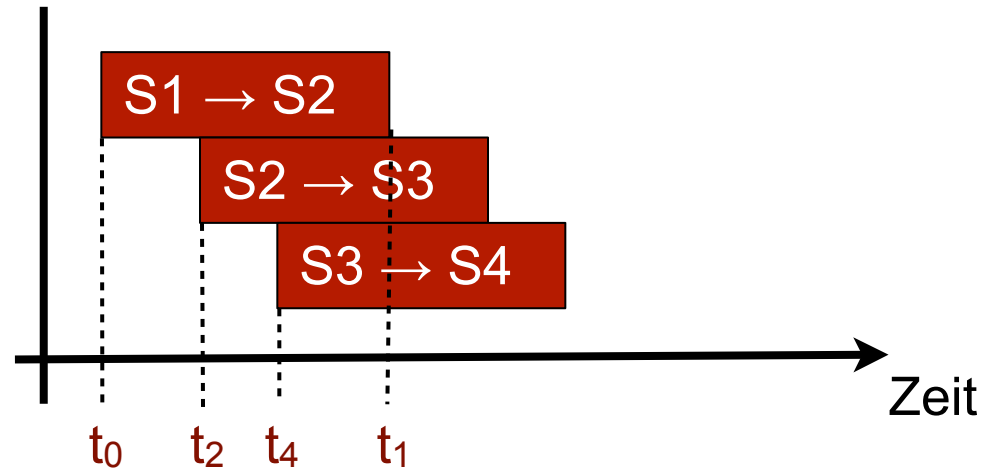
→ Motivation

→ Netzwerkgraph

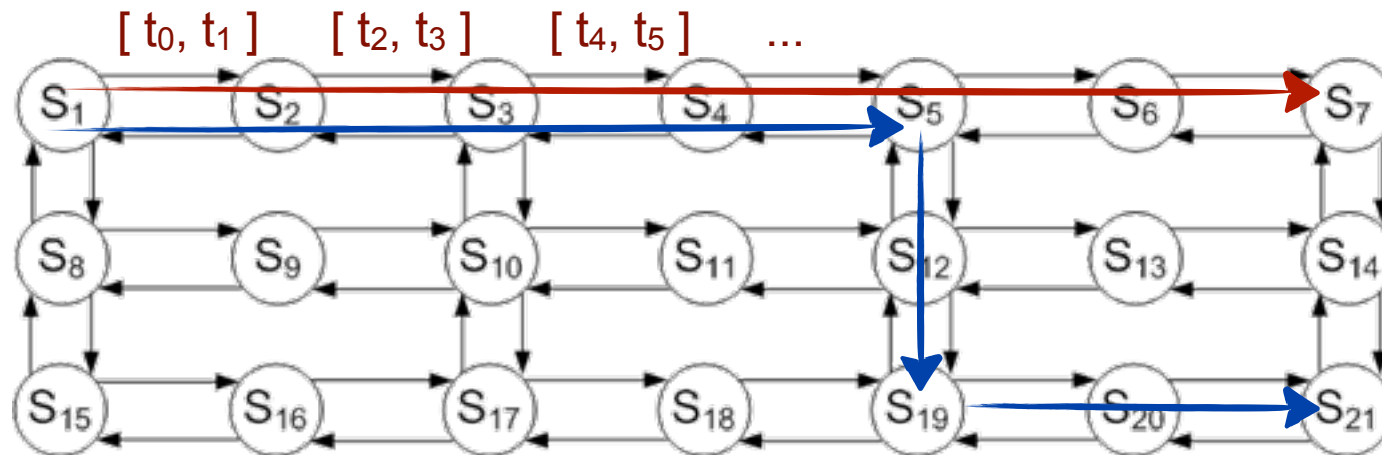
→ Schedule-Qualität

→ Planung

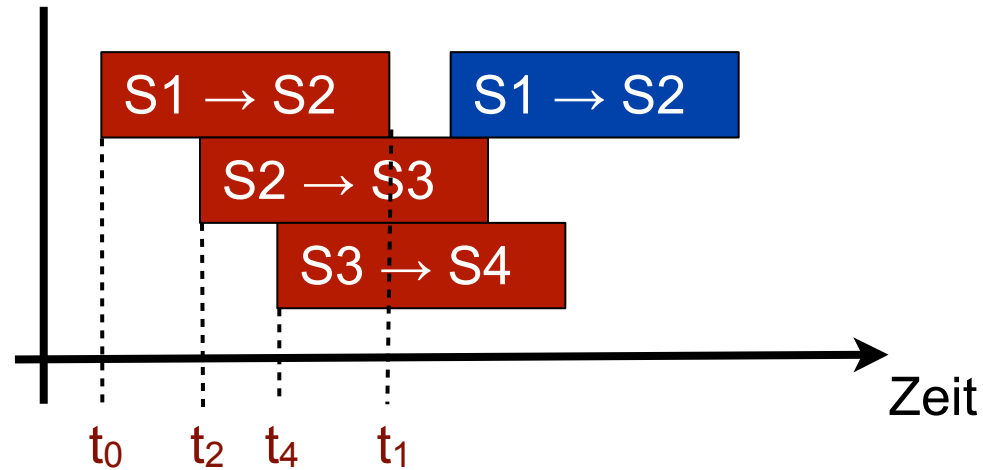
→ Ergebnisse



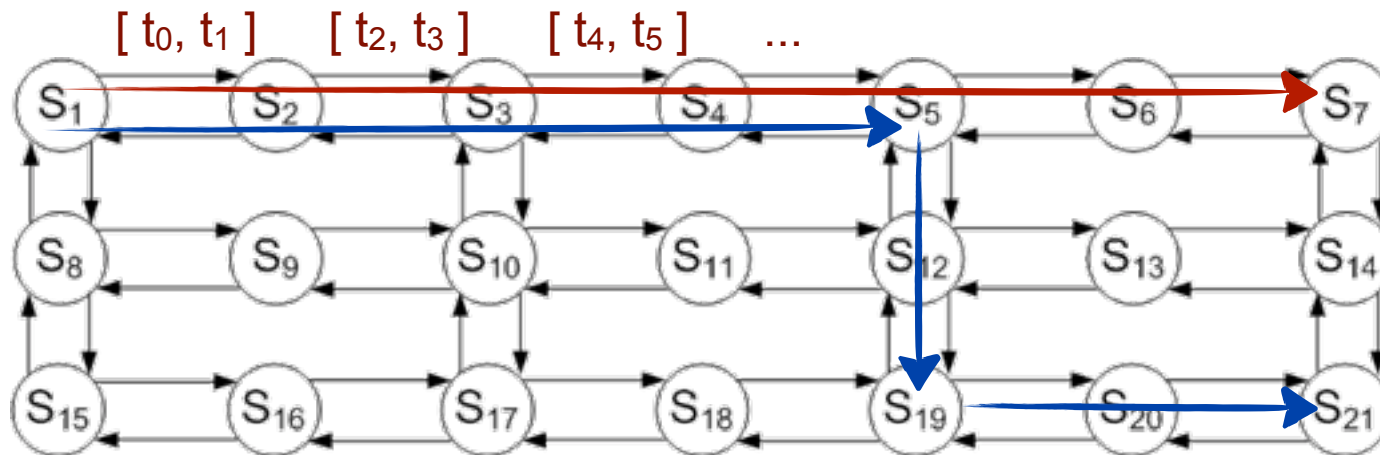
$$t_2 \leq t_1$$



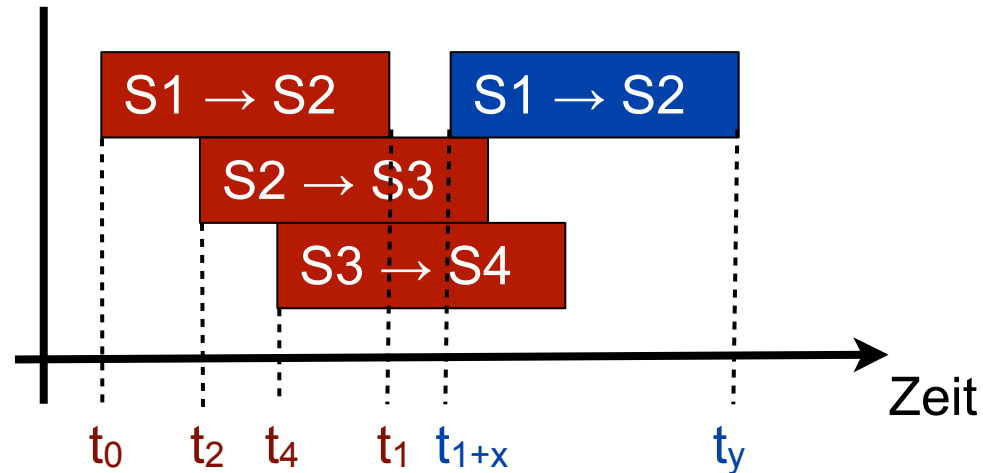
Cut-Through Switching



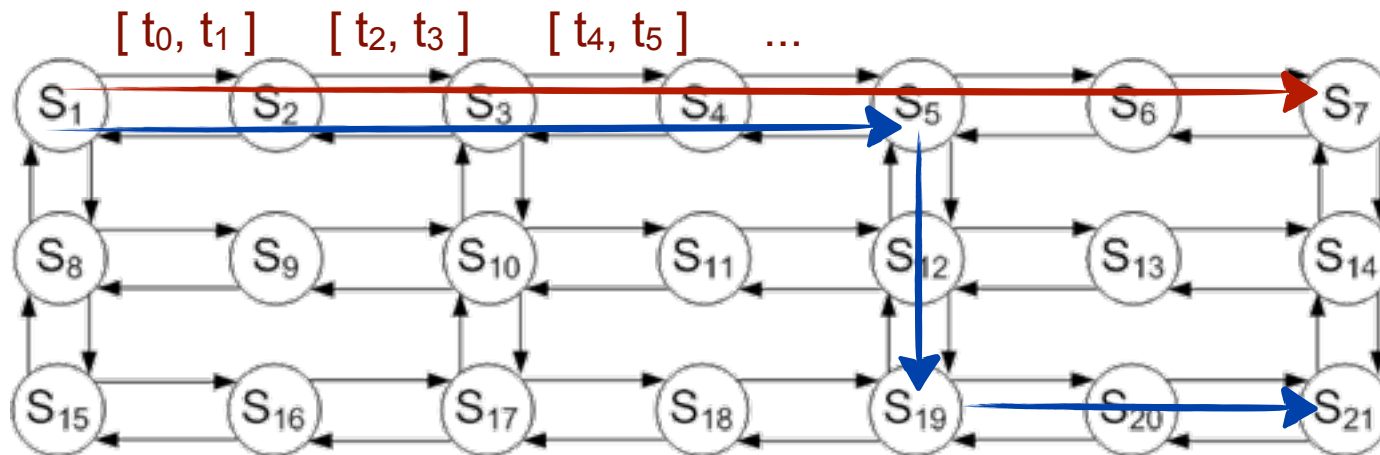
$$t_2 \leq t_1$$



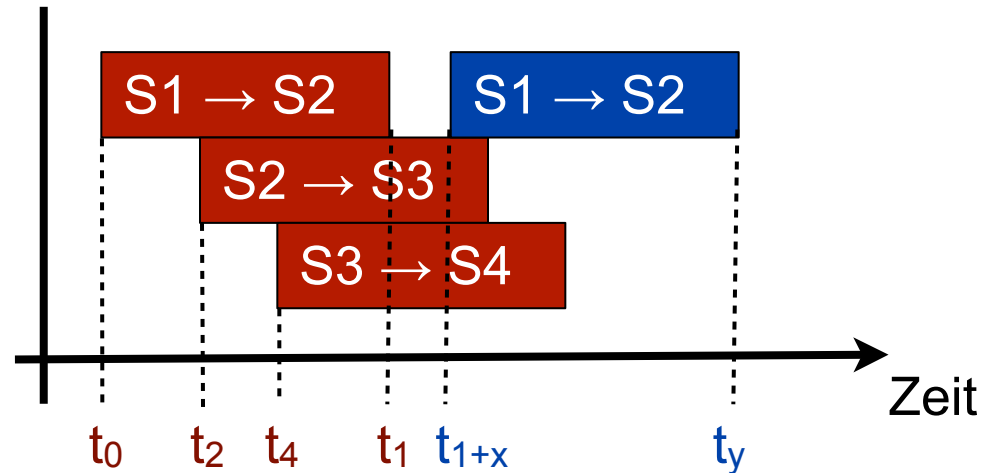
Cut-Through Switching



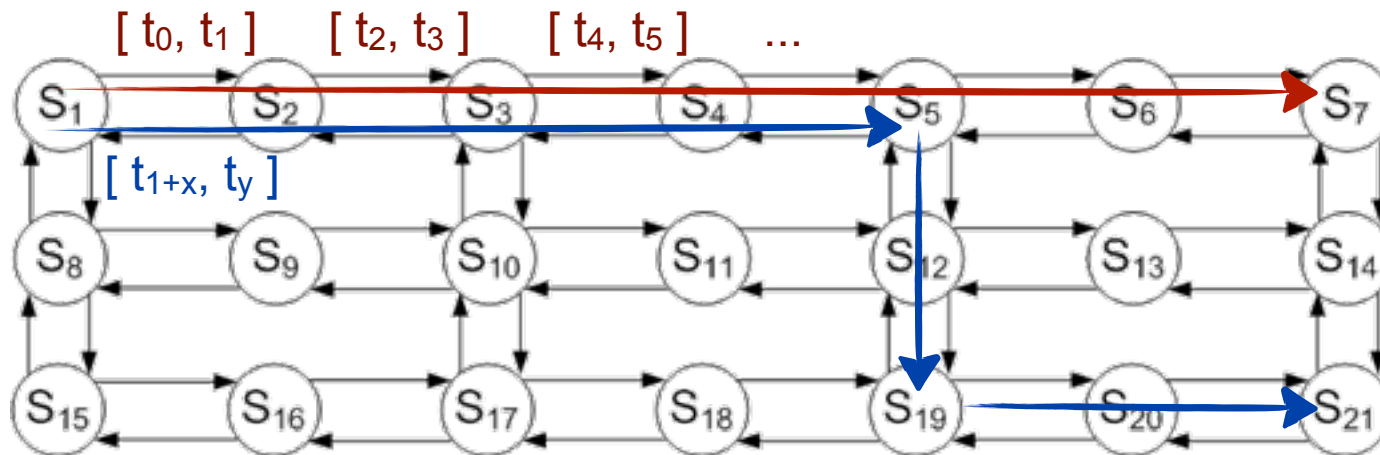
$$t_2 \leq t_1$$



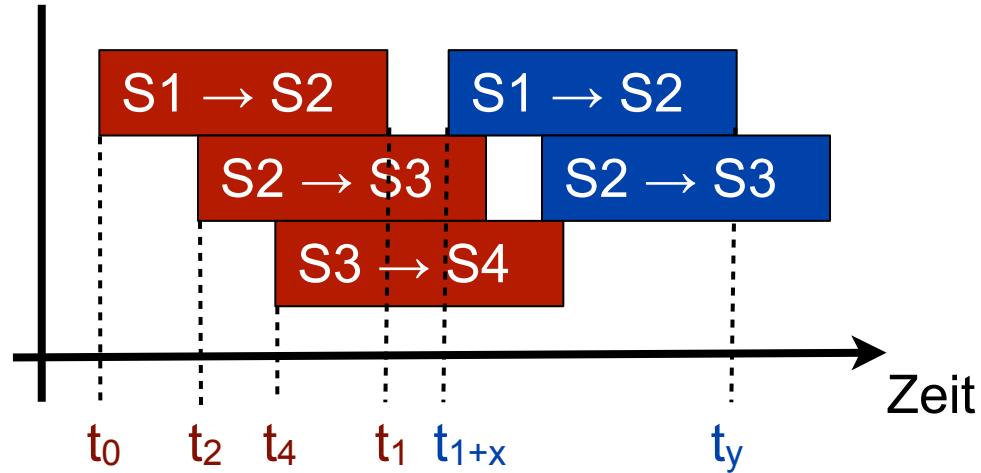
Cut-Through Switching



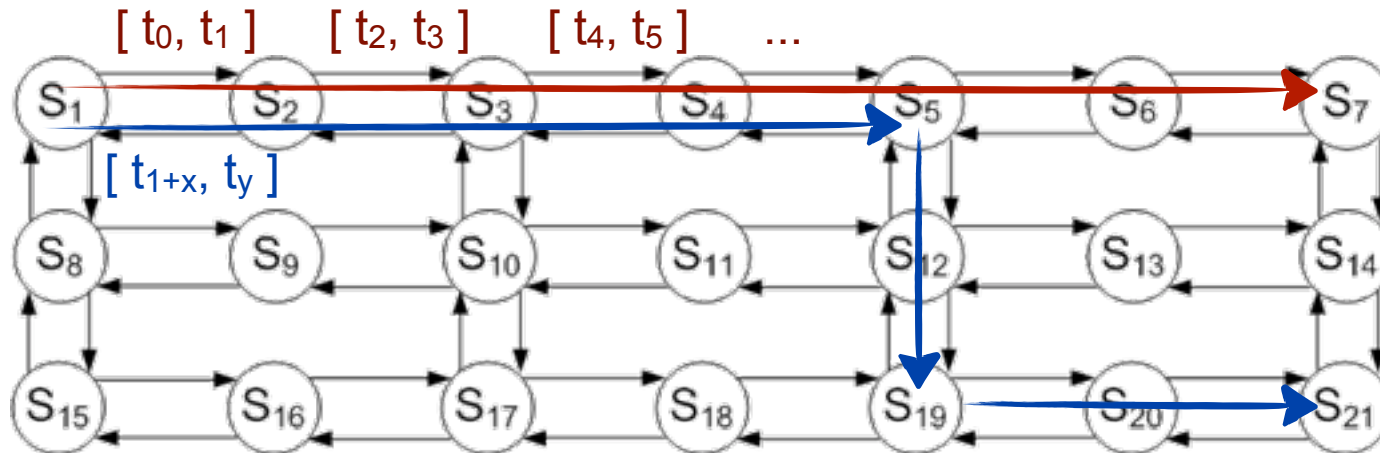
$$t_2 \leq t_1$$



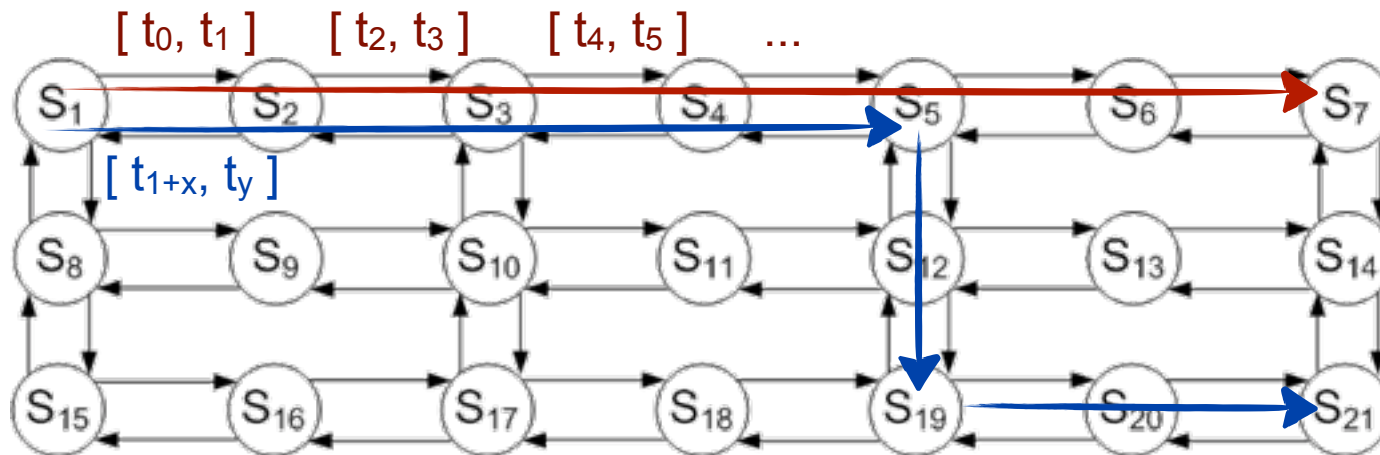
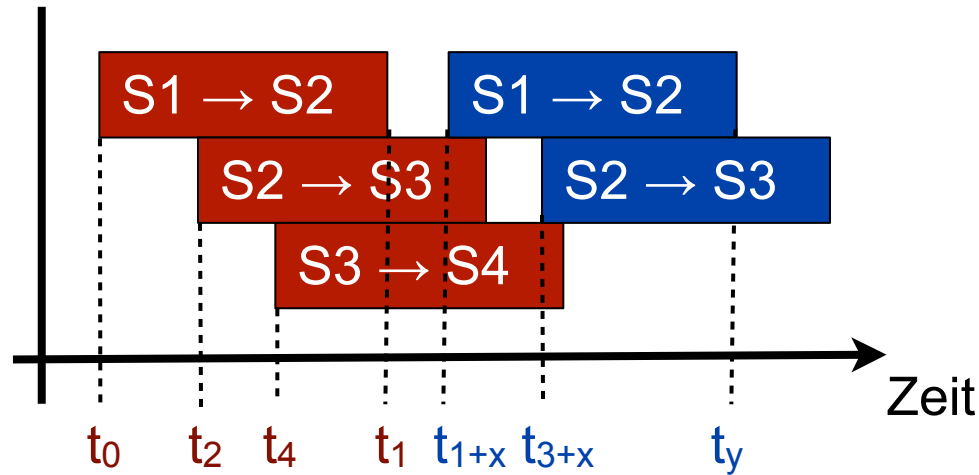
Cut-Through Switching



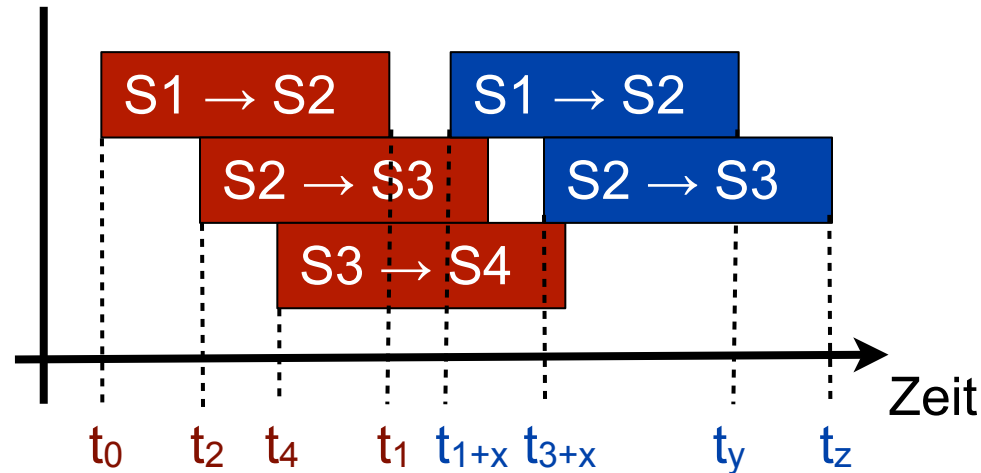
$$t_2 \leq t_1$$



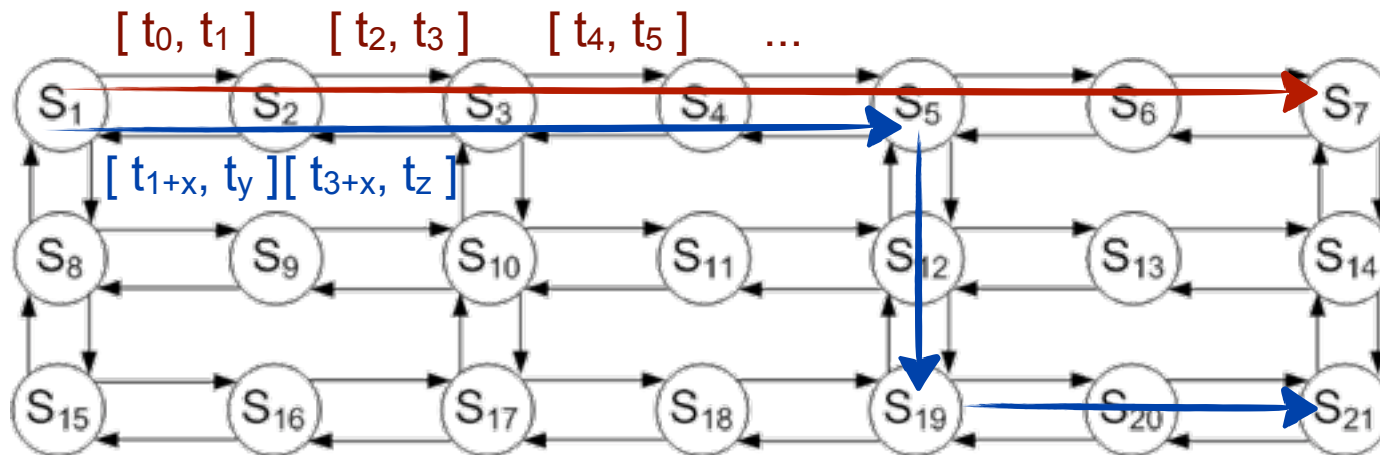
Cut-Through Switching



Cut-Through Switching



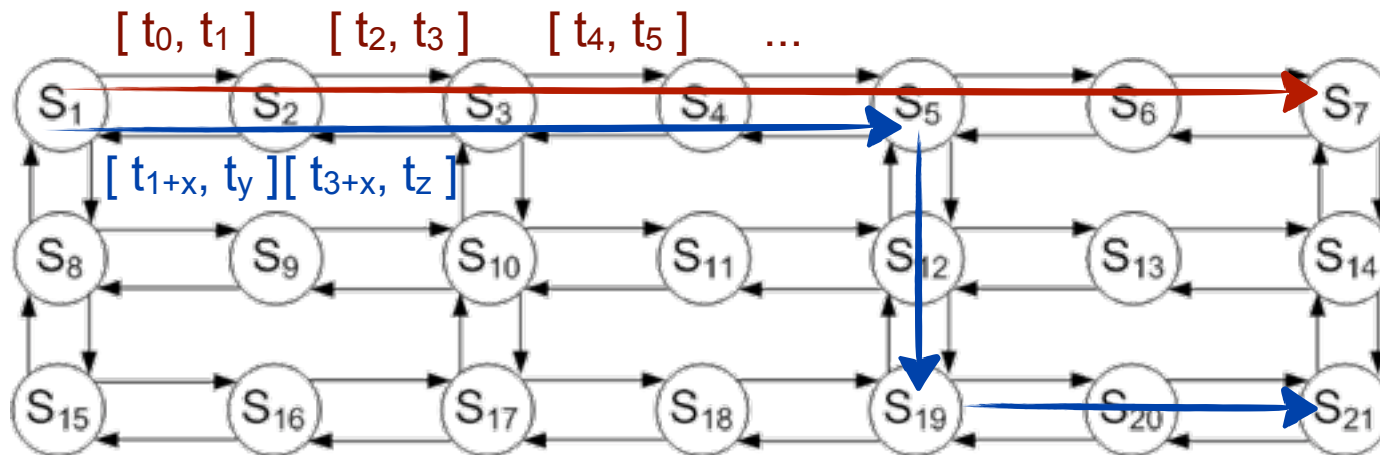
$$t_2 \leq t_1$$



Das Netzwerk als Graph

→ Motivation → **Netzwerkgraph** → Schedule-Qualität → Planung → Ergebnisse

(zum Beispiel m_1 , m_2)



Das Netzwerk als Graph

→ Motivation

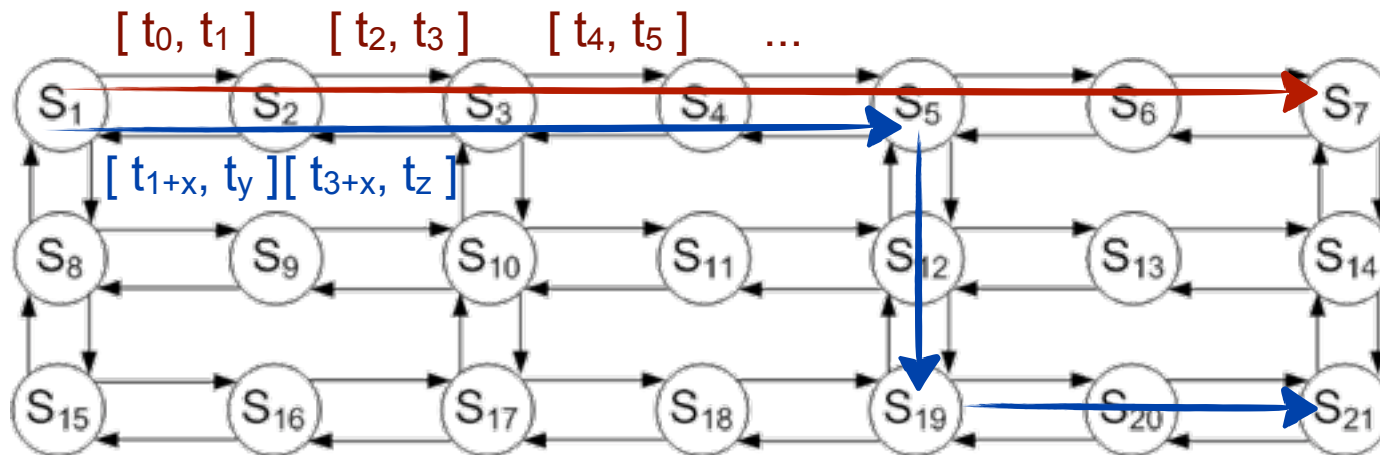
→ Netzwerkgraph

→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$
 - Kantenbelegungsfunktion $\delta_i : p_i \rightarrow \mathbf{R}^2$



Das Netzwerk als Graph

→ Motivation

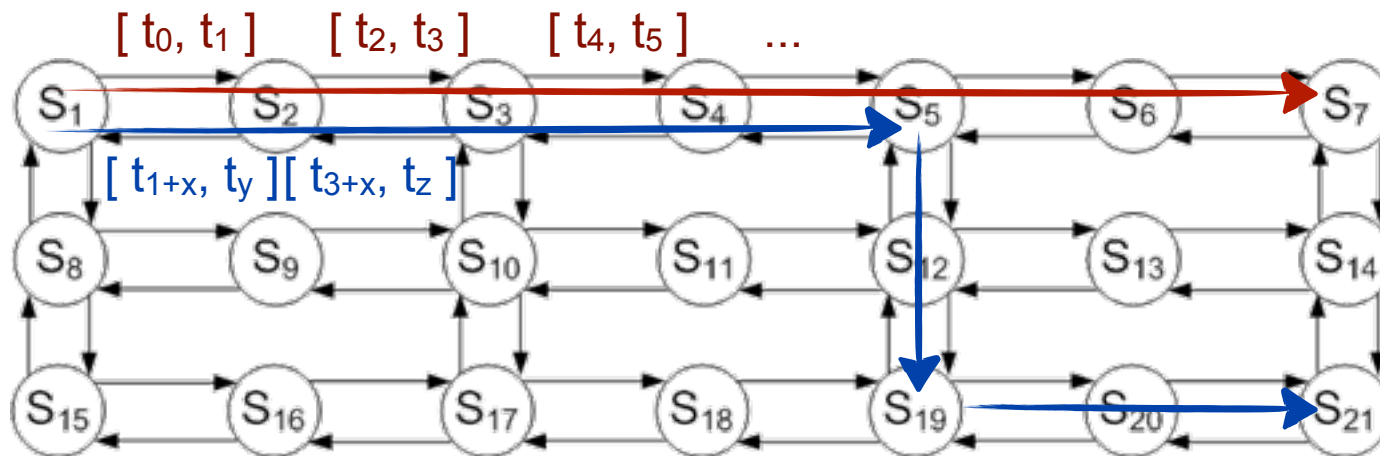
→ Netzwerkgraph

→ Schedule-Qualität

→ Planung

→ Ergebnisse

- Netzwerk kann als gerichteter und gewichteter Graph angesehen werden
- Ein Schedule ist dann gültig, wenn
 - $\forall m_i \in \text{Nachrichten}$ (zum Beispiel m_1, m_2)
 - $\exists p_i \in \text{Pfade}$ zwischen $m_{i, \text{Quelle}}$ und $m_{i, \text{Senke}}$
 - Kantenbelegungsfunktion $\delta_i : p_i \rightarrow \mathbf{R}^2$
 - Bedingung: $t_1 \leq t_{1+x} \wedge t_3 \leq t_{3+x} \wedge \dots$



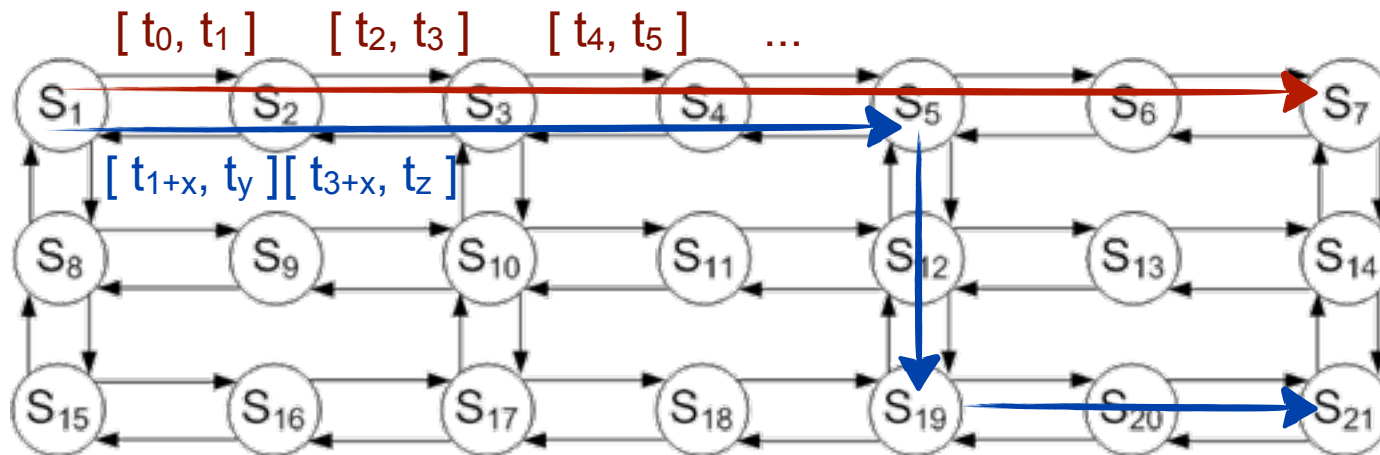
Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

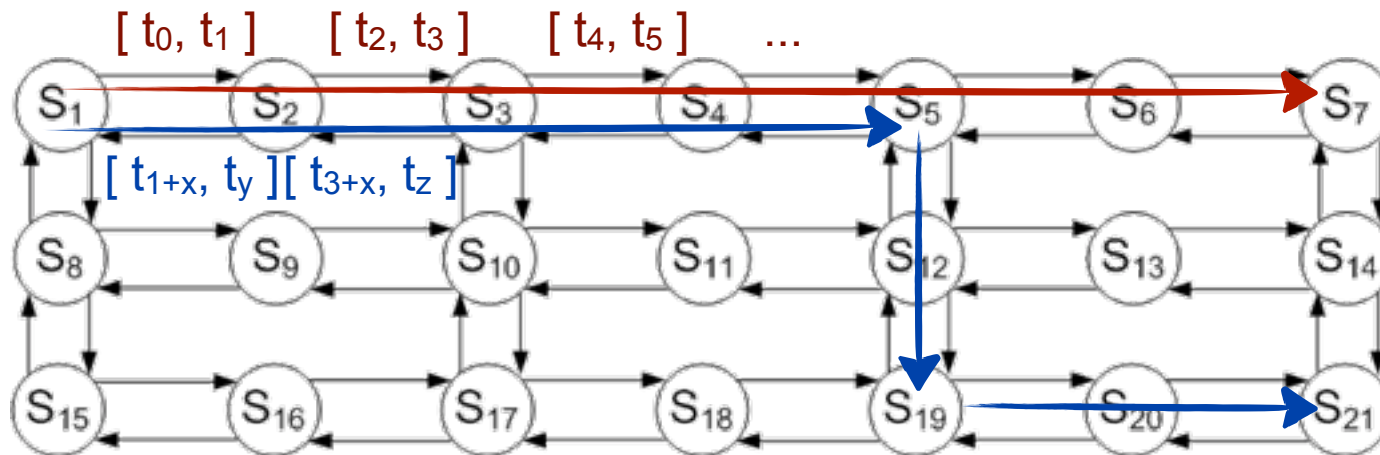
- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.



Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)



Qualität eines Schedules

→ Motivation

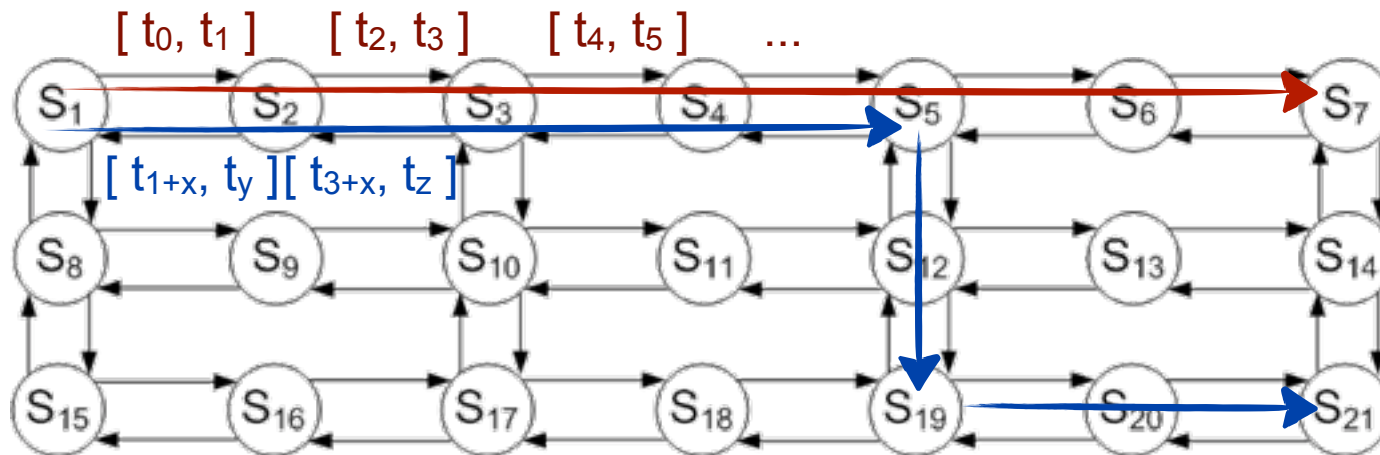
→ Netzwerkgraph

→ **Schedule-Qualität**

→ Planung

→ Ergebnisse

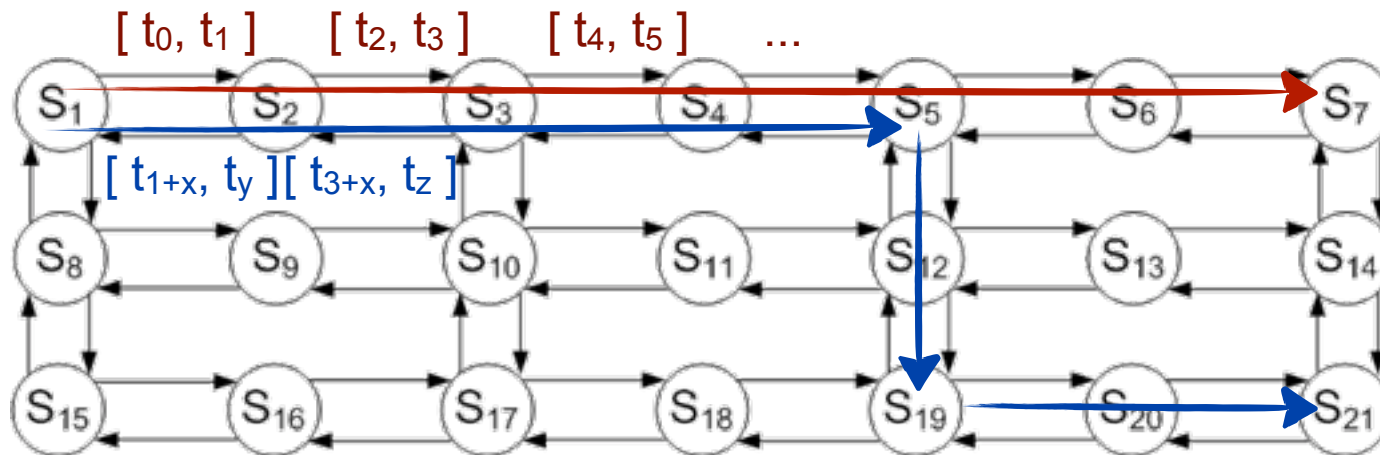
- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**



Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.



Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.
- $\mu(S_{perfect})$ sei ein Schedule in einem perfekten Netzwerk

Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.
- $\mu(S_{perfect})$ sei ein Schedule in einem perfekten Netzwerk
 - keine Kollisionen

Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.
- $\mu(S_{perfect})$ sei ein Schedule in einem perfekten Netzwerk
 - keine Kollisionen
 - keine Delays

Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.
- $\mu(S_{perfect})$ sei ein Schedule in einem perfekten Netzwerk
 - keine Kollisionen
 - keine Delays
 - Alle Nachrichten können sofort versendet werden

Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

- In einem Schedule S wird jede Kante des Netzwerkgraphen für eine Nachricht i für ein bestimmtes Zeitintervall belegt.
- Die Summe aller Intervalllängen (Belegungen) beschreibt die Qualität eines Schedules (S)
 - **Qualität eines Schedules $S = \mu(S)$**
 - Je geringer $\mu(S)$, desto besser.
- $\mu(S_{perfect})$ sei ein Schedule in einem perfekten Netzwerk
 - keine Kollisionen
 - keine Delays
 - Alle Nachrichten können sofort versendet werden
 - Übertragungsdauer hängt nur ab von
 - Nachrichtenlänge
 - Übertragungsrate eines Netzwerkes

Relative Qualität eines Schedules

→ Motivation

→ Netzwerkgraph

→ **Schedule-Qualität**

→ Planung

→ Ergebnisse

$$v(S_{\text{real}}) = \frac{\mu(S_{\text{perfect}})}{\mu(S_{\text{real}})}$$

Relative Qualität eines Schedules

→ Motivation → Netzwerkgraph → **Schedule-Qualität** → Planung → Ergebnisse

Zur Bewertung eines Schedules wird (hier) die relative Qualität eines Schedules herangezogen:

$$v(S_{\text{real}}) = \frac{\mu(S_{\text{perfect}})}{\mu(S_{\text{real}})}$$

Relative Qualität eines Schedules

→ Motivation

→ Netzwerkgraph

→ **Schedule-Qualität**

→ Planung

→ Ergebnisse

Zur Bewertung eines Schedules wird (hier) die relative Qualität eines Schedules herangezogen:

$$v(S_{\text{real}}) = \frac{\mu(S_{\text{perfect}})}{\mu(S_{\text{real}})}$$

Die relative Qualität liegt in einem Bereich zwischen $0 < v(S_{\text{real}}) \leq 1$, mit $v(S_{\text{real}}) = 1$ für einen perfekten Schedule in einem perfekten Netzwerk.

Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

- Vollständige Suche
 - Erzeugung aller gültigen Schedules

Durchführung der Planung

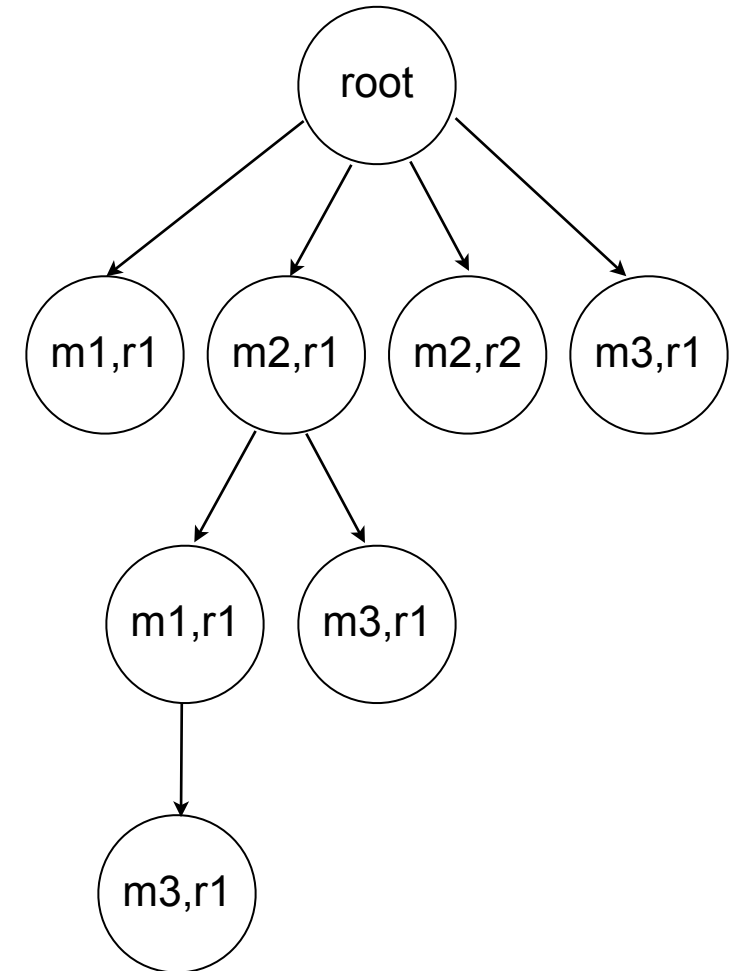
→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic

Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

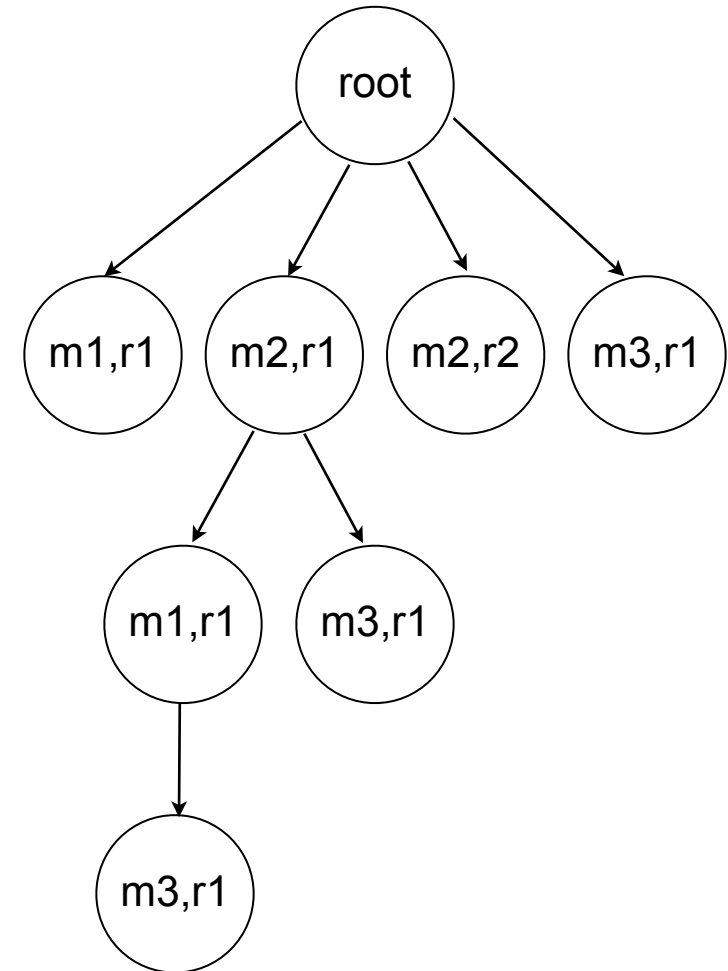
- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic
 - Baum als Suchraum



Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

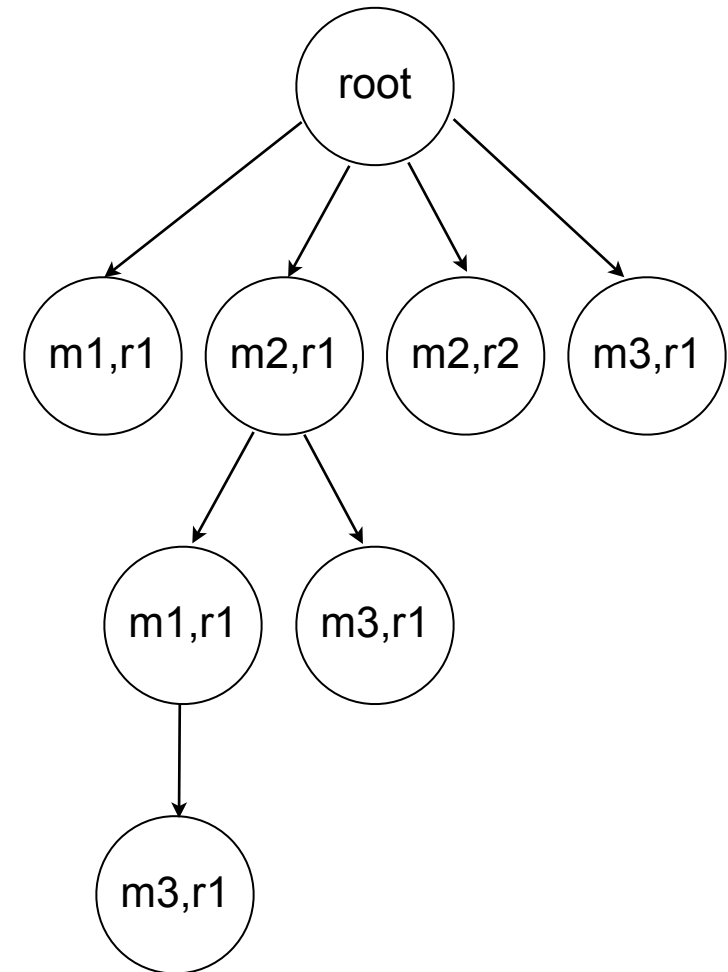
- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic
 - Baum als Suchraum
 - Konstruiert eine Lösung



Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

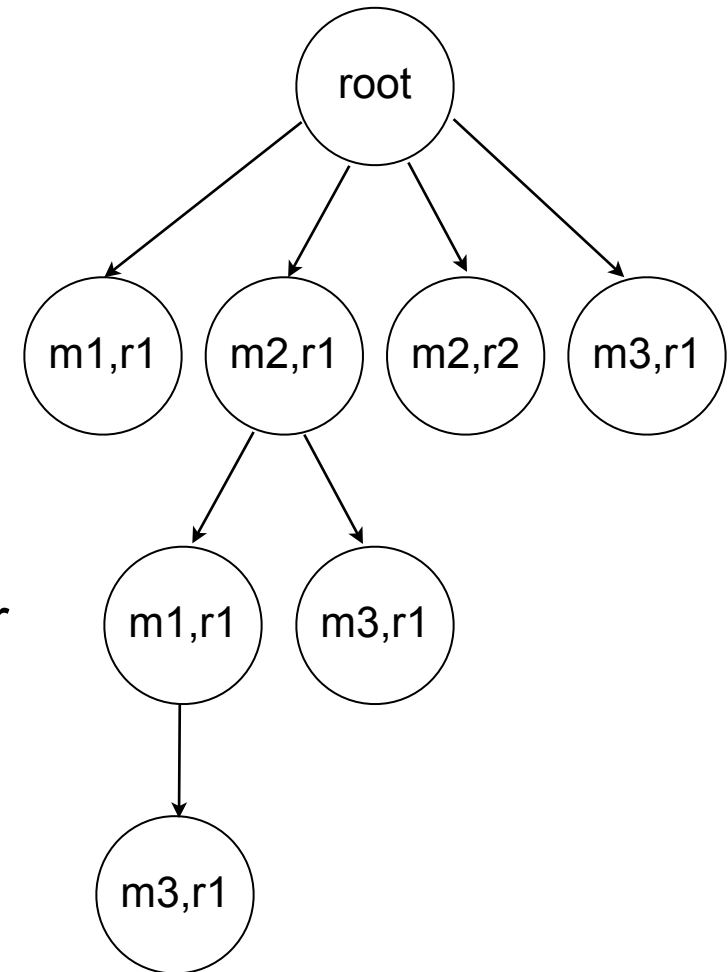
- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic
 - Baum als Suchraum
 - Konstruiert eine Lösung
 - 2 Bewertungsfunktionen in jedem Erweiterungsschritt



Durchführung der Planung

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

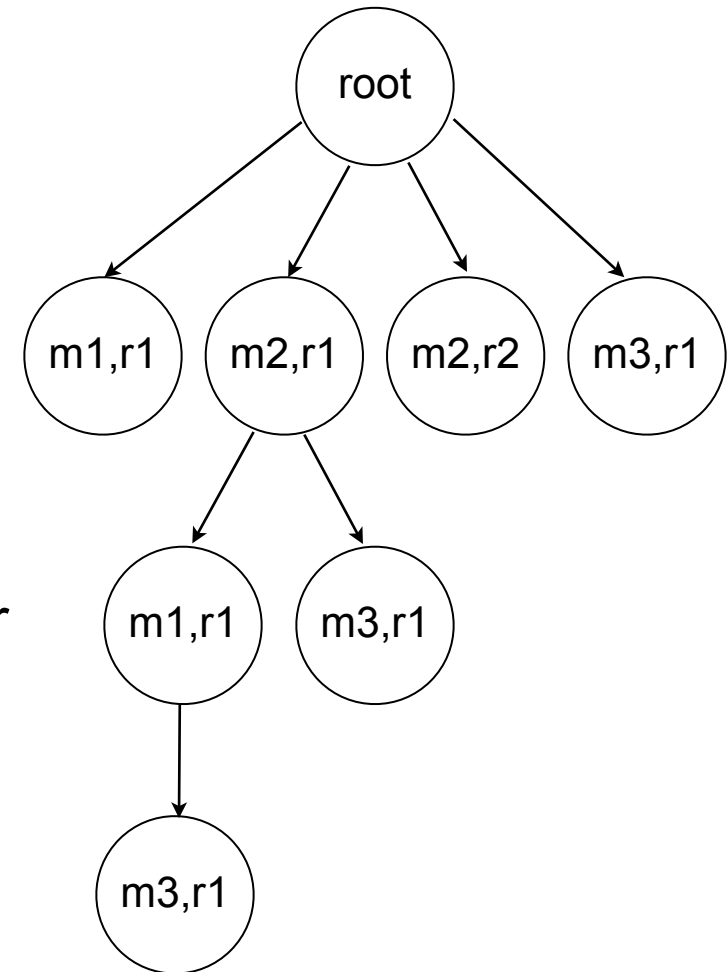
- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic
 - Baum als Suchraum
 - Konstruiert eine Lösung
 - 2 Bewertungsfunktionen in jedem Erweiterungsschritt
 - Evaluierung: $f(m, r)$
 - Summe aller Kantengewichte bisher (Netzwerkgraph)



Durchführung der Planung

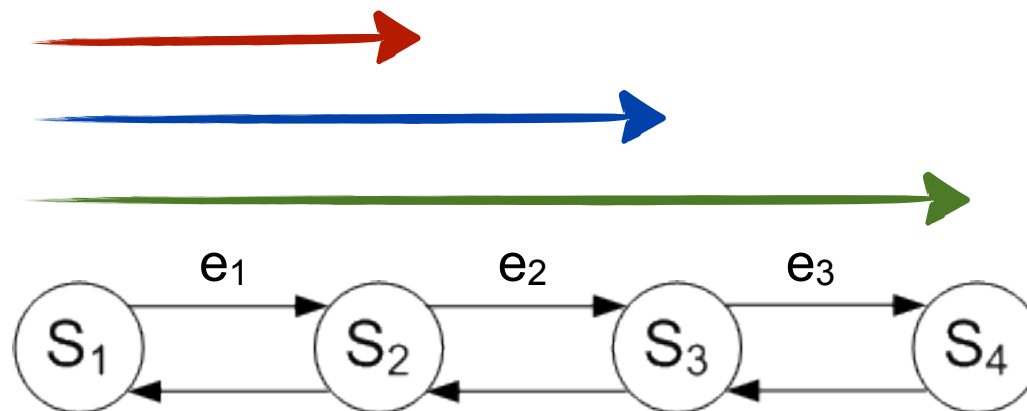
→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

- Vollständige Suche
 - Erzeugung aller gültigen Schedules
- Greedy Best-First Search Heuristic
 - Baum als Suchraum
 - Konstruiert eine Lösung
 - 2 Bewertungsfunktionen in jedem Erweiterungsschritt
 - Evaluierung: $f(m, r)$
 - Summe aller Kantengewichte bisher (Netzwerkgraph)
 - Heuristische Bewertung: $h(m, r, e)$
 - Bottleneck-Prediction



Bottleneck Prediction

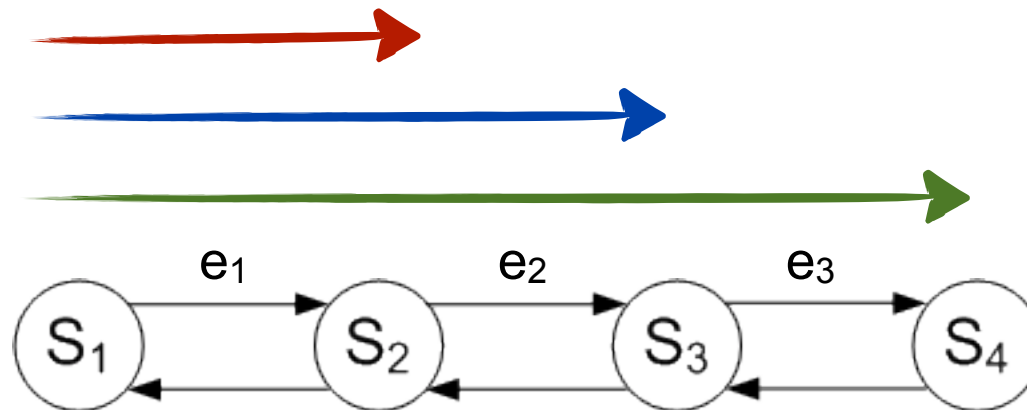
→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse



Bottleneck Prediction

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

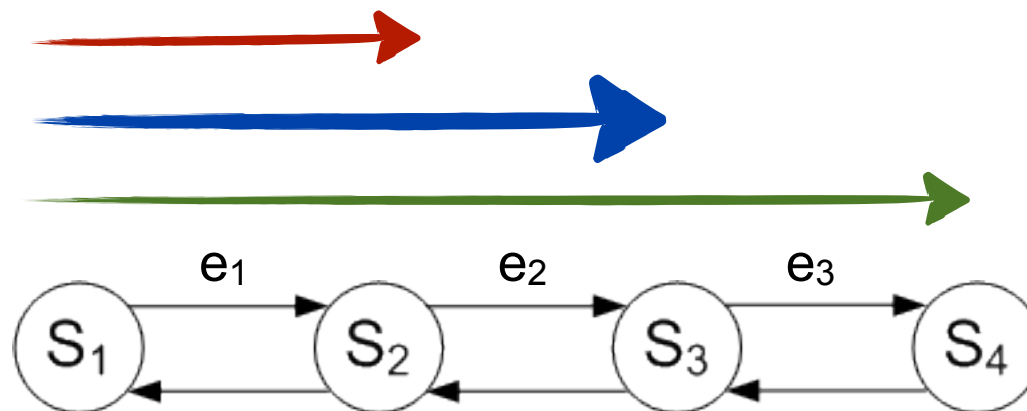
1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)



Bottleneck Prediction

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

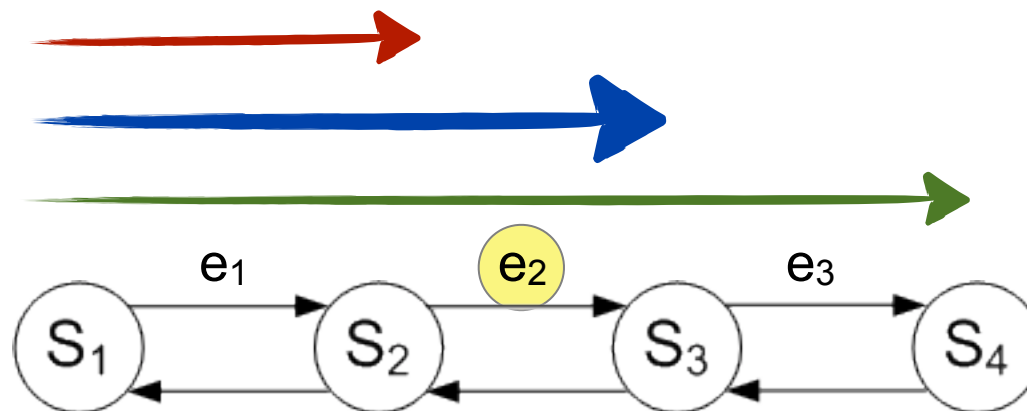
1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)
2. Annahme: m_2 wird zuerst geplant



Bottleneck Prediction

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → Ergebnisse

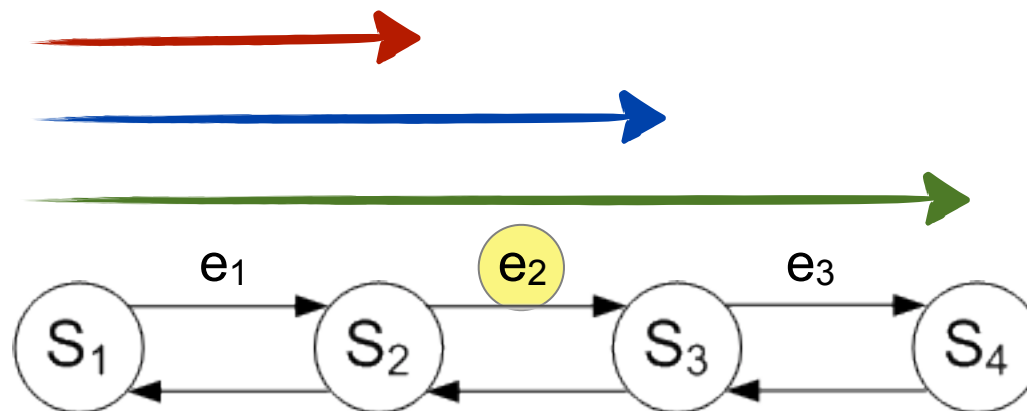
1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)
2. Annahme: m_2 wird zuerst geplant
→ e_2 Bottleneck Edge



Bottleneck Prediction

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)
2. Annahme: m_2 wird zuerst geplant
 - e_2 Bottleneck Edge
 - Ausstehende Nachrichten: m_1 , m_3

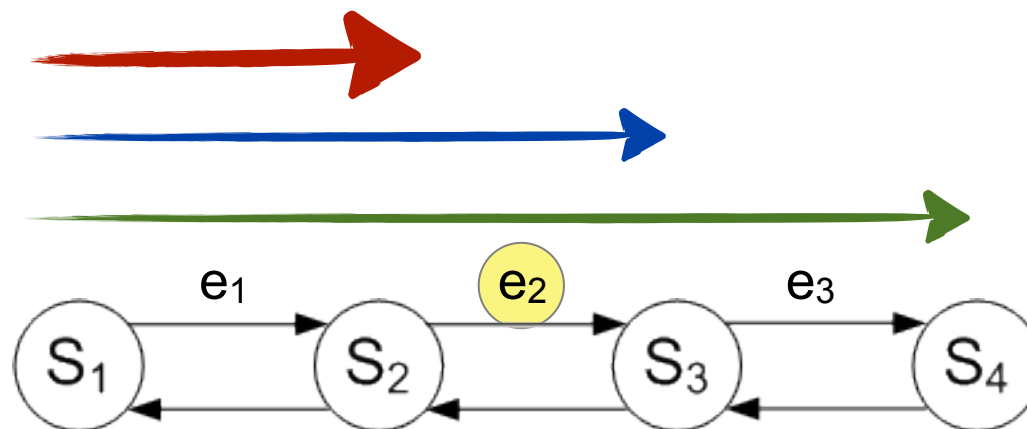


Bottleneck Prediction

1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)
2. Annahme: m_2 wird zuerst geplant
 - e_2 Bottleneck Edge
 - Ausstehende Nachrichten: m_1 , m_3

Prediction für m_1 :

m_1 wird nicht über e_2 gehen → Prediction(m_1) = 0



Bottleneck Prediction

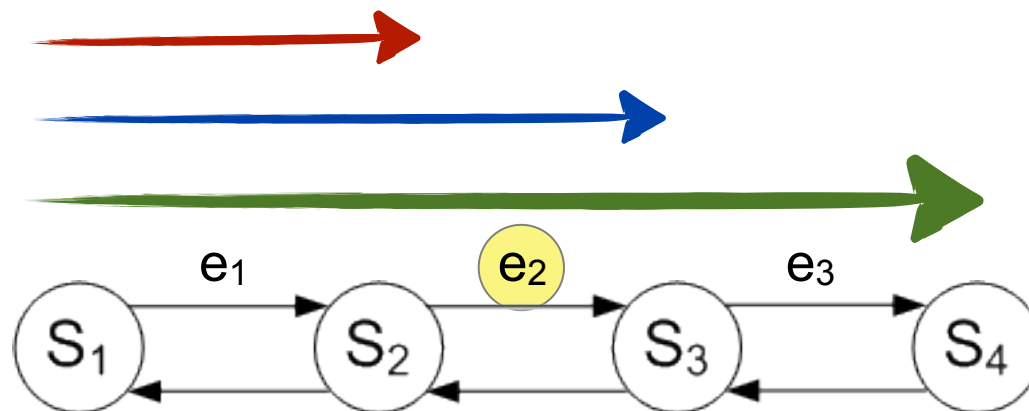
1. Annahme: 3 Nachrichten (m_1 , m_2 , m_3)
2. Annahme: m_2 wird zuerst geplant
 - e_2 Bottleneck Edge
 - Ausstehende Nachrichten: m_1 , m_3

Prediction für m_1 :

m_1 wird nicht über e_2 gehen → $\text{Prediction}(m_1) = 0$

Prediction für m_3 :

m_3 wird über e_2 gehen → $\text{Prediction}(m_3) =$ Übertragungsdauer von m_3 über e_2



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

→ Schedule-Qualität

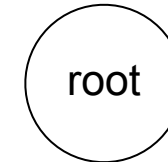
→ Planung

→ Ergebnisse

Greedy Best-First Search

→ Motivation → Netzwerkgraph → Schedule-Qualität → **Planung** → Ergebnisse

1. Beginne mit leerem Startknoten



Greedy Best-First Search

→ Motivation

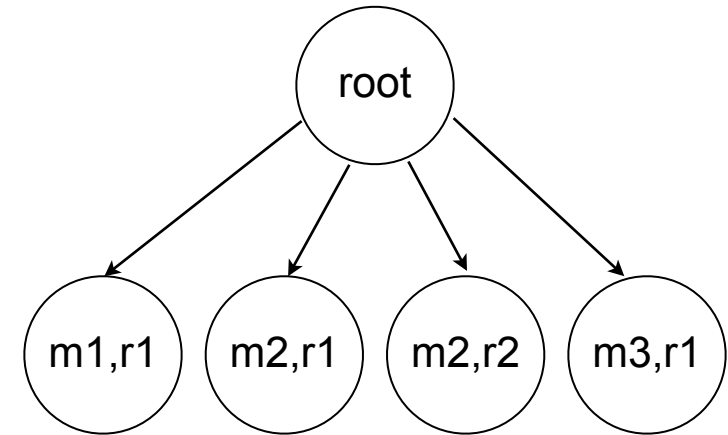
→ Netzwerkgraph

→ Schedule-Qualität

→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

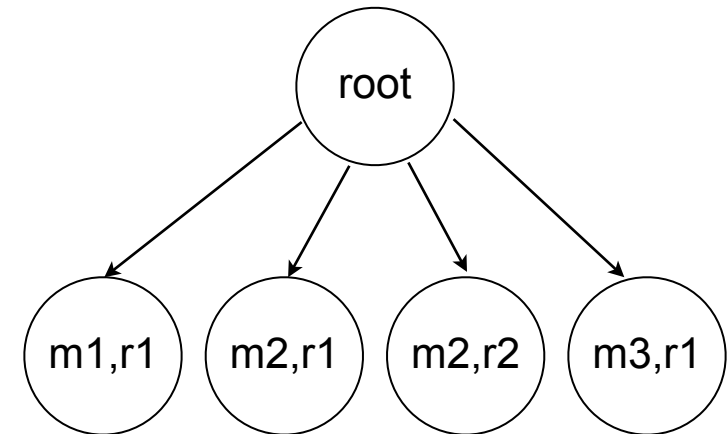
→ Schedule-Qualität

→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein
3. Wähle und erweitere den Knoten mit der besten Bewertung:

$$\min(f(m, r) + h(m, r, e))$$



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

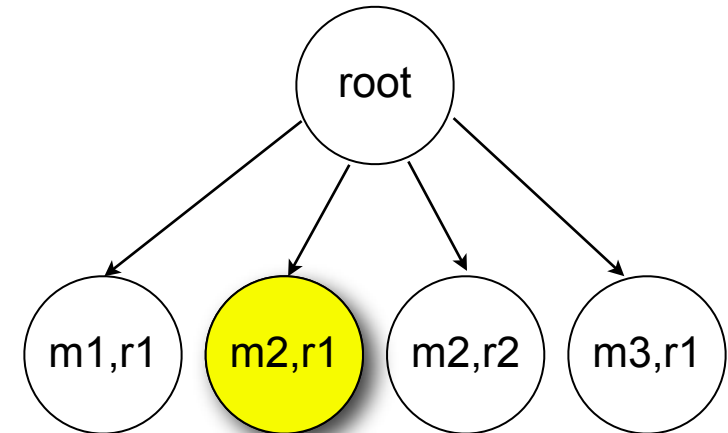
→ Schedule-Qualität

→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein
3. Wähle und erweitere den Knoten mit der besten Bewertung:

$$\min(f(m, r) + h(m, r, e))$$



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

→ Schedule-Qualität

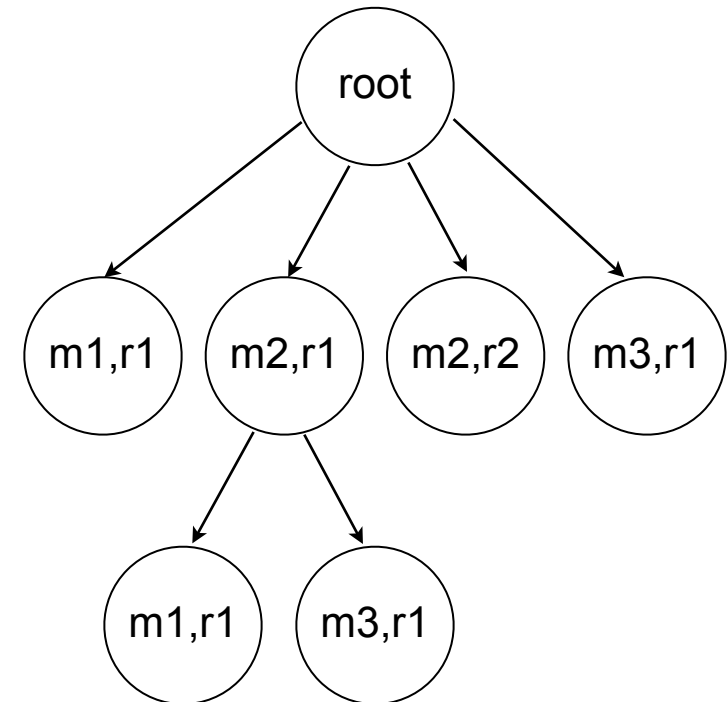
→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein
3. Wähle und erweitere den Knoten mit der besten Bewertung:

$$\min(f(m, r) + h(m, r, e))$$

4. Erweitere den gewählten Knoten mit den verbliebenen Kombinationen (Nachrichten & Routen)



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

→ Schedule-Qualität

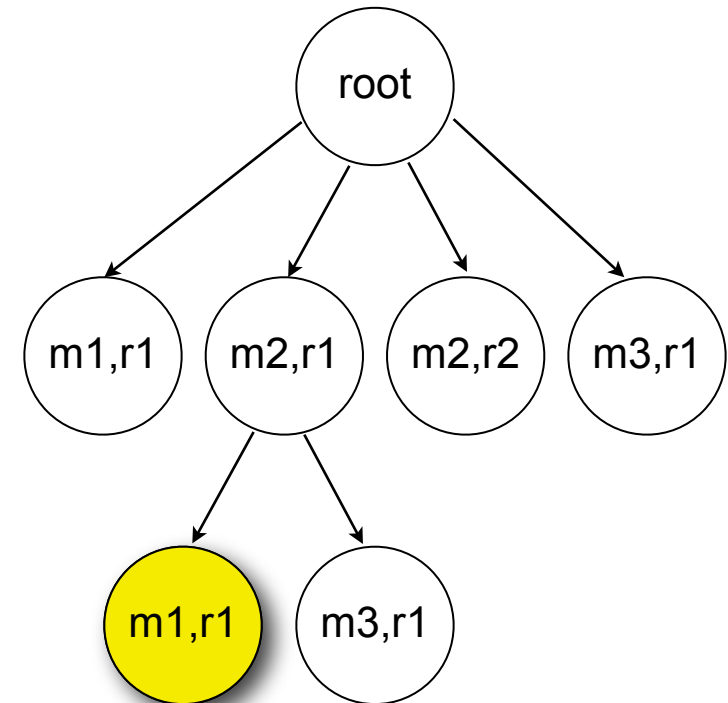
→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein
3. Wähle und erweitere den Knoten mit der besten Bewertung:

$$\min(f(m, r) + h(m, r, e))$$

4. Erweitere den gewählten Knoten mit den verbliebenen Kombinationen (Nachrichten & Routen)
5. Ende, falls nur noch ein oder gar kein Knoten eingehangen wurde, sonst Schritt 3



Greedy Best-First Search

→ Motivation

→ Netzwerkgraph

→ Schedule-Qualität

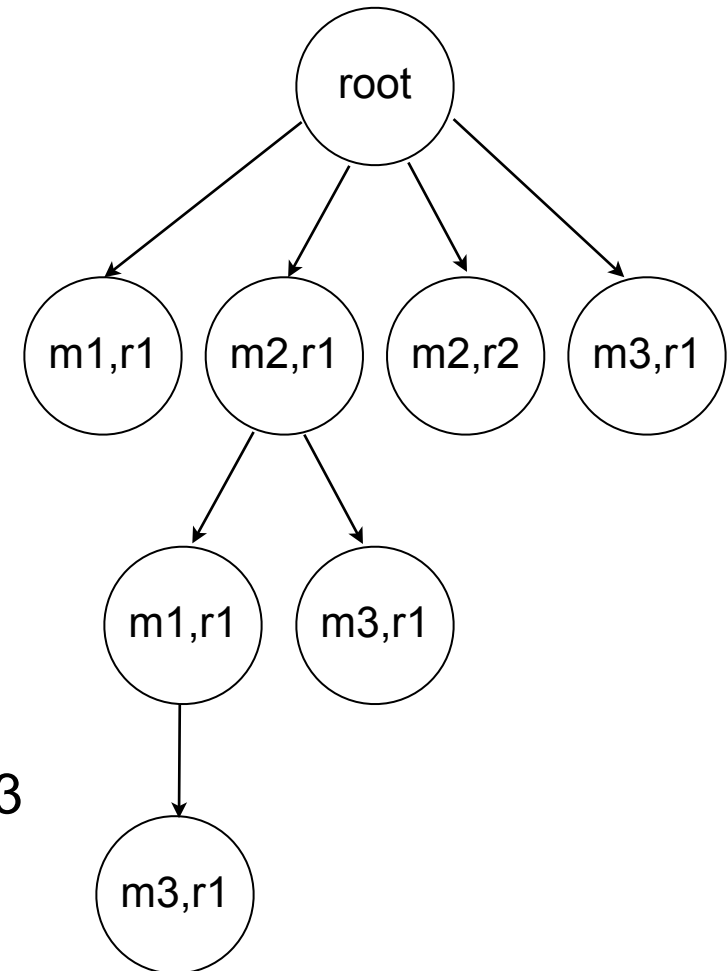
→ Planung

→ Ergebnisse

1. Beginne mit leerem Startknoten
2. Hängen alle Kombinationen von Nachrichten (m) & Routen (r) ein
3. Wähle und erweitere den Knoten mit der besten Bewertung:

$$\min(f(m, r) + h(m, r, e))$$

4. Erweitere den gewählten Knoten mit den verbliebenen Kombinationen (Nachrichten & Routen)
5. Ende, falls nur noch ein oder gar kein Knoten eingehangen wurde, sonst Schritt 3



Die Ergebnisse

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Die Ergebnisse

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Untersucht wurde

- die Aufrufhäufigkeit der Funktion, die für die Erzeugung der δ_i eines Schedules zuständig ist

Die Ergebnisse

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Untersucht wurde

- die Aufrufhäufigkeit der Funktion, die für die Erzeugung der δ_i eines Schedules zuständig ist
 - Komplexeste Funktion
 - Wird in jedem Iterationsschritt aufgerufen

Die Ergebnisse

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Untersucht wurde

- die Aufrufhäufigkeit der Funktion, die für die Erzeugung der δ_i eines Schedules zuständig ist
 - Komplexeste Funktion
 - Wird in jedem Iterationsschritt aufgerufen
- die relative Qualität der Schedules
 - Vergleichsmöglichkeit

Die Ergebnisse

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Untersucht wurde

- die Aufrufhäufigkeit der Funktion, die für die Erzeugung der δ_i eines Schedules zuständig ist
 - Komplexeste Funktion
 - Wird in jedem Iterationsschritt aufgerufen
- die relative Qualität der Schedules
 - Vergleichsmöglichkeit

Szenario

- 1 Programmable Logic Controller (PLC)
- n IO-Devices
- Pro Zyklus sendet jedes IO-Device eine Nachricht an die PLC
- Pro Zyklus empfängt jedes IO-Device eine Nachricht von der PLC

Die Ergebnisse - Die Linie

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Anzahl der Teilnehmer	Vollständige Suche Iterationsschritte	Greedy BFS Iterationsschritte
1+2	96 (0.658)	10 (0.658)
1+3	4320 (0.457)	21 (0.457)
1+4	322560 (0.368)	36 (0.368)
1+5	-	55 (0.317)
1+6	-	78 (0.264)

Die Ergebnisse - Die Linie

→ Motivation → Netzwerkgraph → Schedule-Qualität → Planung → **Ergebnisse**

Anzahl der Teilnehmer	Vollständige Suche Iterationsschritte	Greedy BFS Iterationsschritte
1+2	96 (0.658)	10 (0.658)
1+3	4320 (0.457)	21 (0.457)
1+4	322560 (0.368)	36 (0.368)
1+5	-	55 (0.317)
1+6	-	78 (0.264)

Greedy Best-First Search:
Gleiche Qualität bei deutlich geringerem Aufwand

Die Ergebnisse - Vermaschte Netze

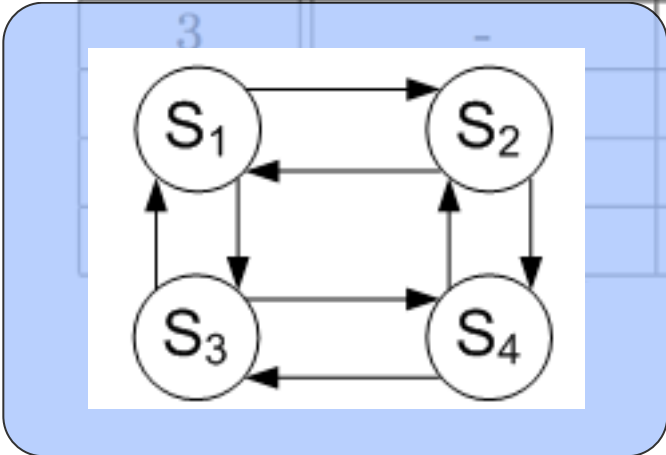
[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)		42 (0.64)		
2	-	-			
3	-	-			
4	-	-	-	-	
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

➔ Motivation
➔ Netzwerkgraph
➔ Schedule-Qualität
➔ Planung
➔ Ergebnisse

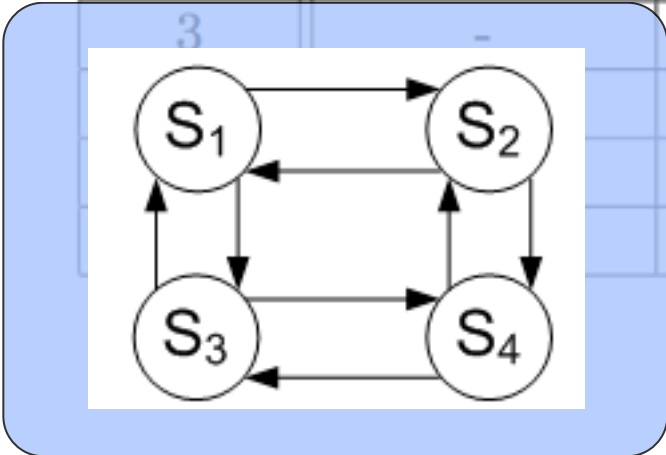
Network type	Complete Search all routes	Search best route	Greedy Best-first search		
			all routes with prediction	without prediction	best route with or without prediction
1	276480 (0.63)		42 (0.64)		
2	-	-			
3	-	-			
		-	-	-	
		-	-	-	
		-	-	-	



Die Ergebnisse - Vermaschte Netze

→ Motivation
→ Netzwerkgraph
→ Schedule-Qualität
→ Planung
→ Ergebnisse

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	with prediction	without prediction	with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-			
3	-	-			
	-	-	-	-	
	-	-	-	-	
	-	-	-	-	



Die Ergebnisse - Vermaschte Netze

[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-			
3	-	-			
4	-	-	-	-	
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

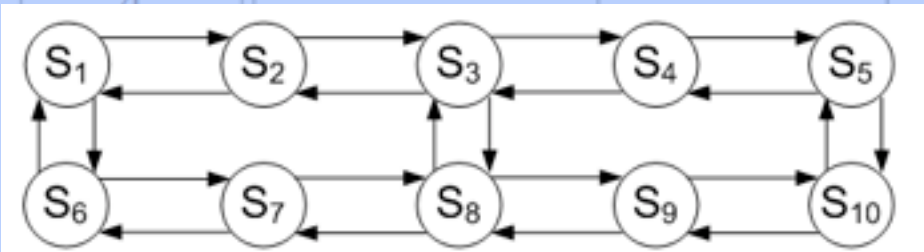
[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-			
4	-	-	-	-	
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

→ Motivation
→ Netzwerkgraph
→ Schedule-Qualität
→ Planung
→ Ergebnisse

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	with prediction	without prediction	with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-			
4			-	-	
			-	-	
			-	-	



Die Ergebnisse - Vermaschte Netze

[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-			
4	-	-	-	-	
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

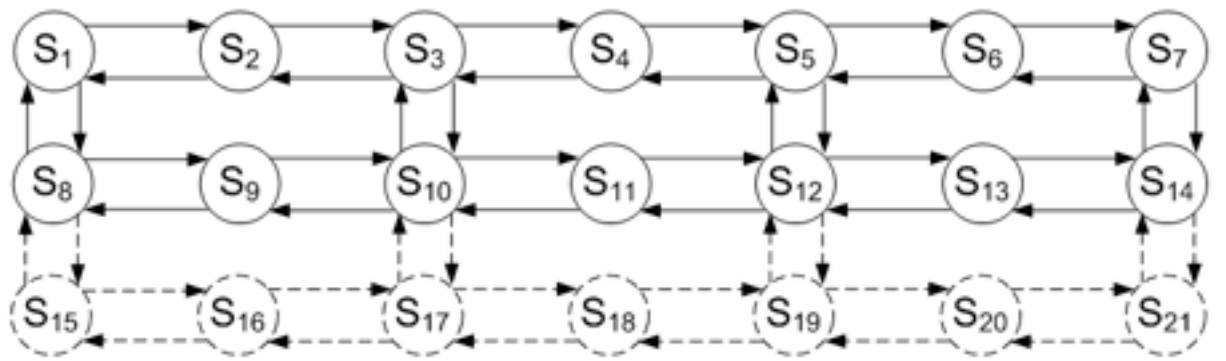
[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
4	-	-	-	-	
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

➔ Motivation
➔ Netzwerkgraph
➔ Schedule-Qualität
➔ Planung
➔ Ergebnisse

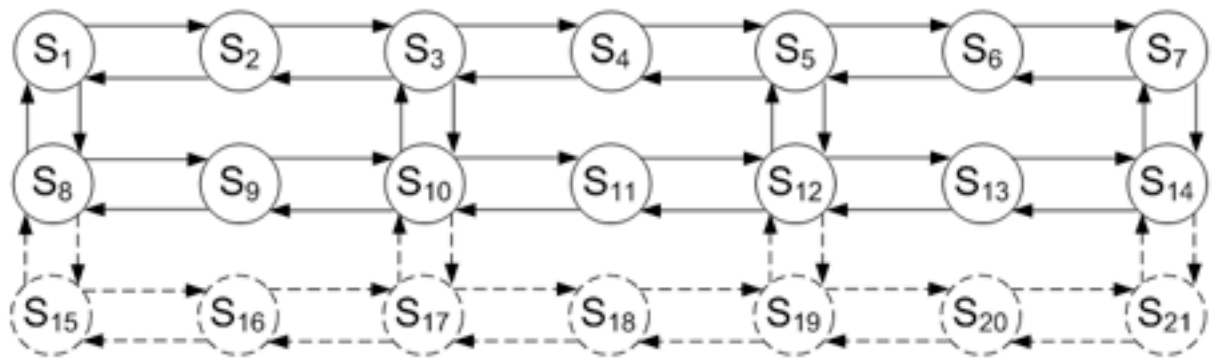
Network type	Complete Search		Greedy Best-first search		
	all routes	best route	with prediction	without prediction	with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
4	-	-	-		
	-	-	-		
	-	-	-		



Die Ergebnisse - Vermaschte Netze

➔ Motivation
➔ Netzwerkgraph
➔ Schedule-Qualität
➔ Planung
➔ Ergebnisse

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	with prediction	without prediction	with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
4	-	-	-		820 (0.13)



Die Ergebnisse - Vermaschte Netze

[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
4	-	-	-	-	820 (0.13)
5	-	-	-	-	
6	-	-	-	-	

Die Ergebnisse - Vermaschte Netze

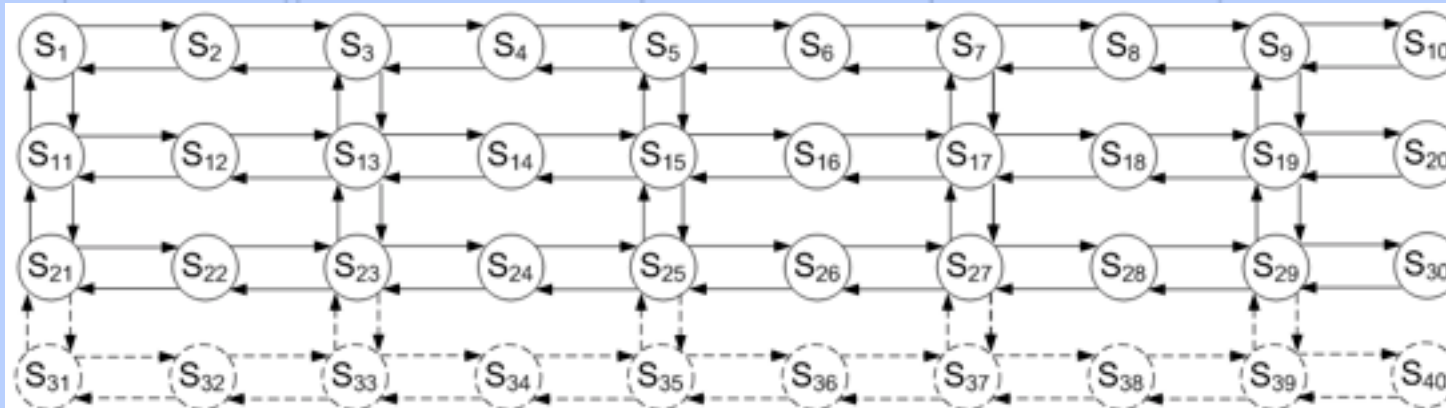
[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
4	-	-	-	-	820 (0.13)
5	-	-	-	-	1711 (0.09)
6	-	-	-	-	3081 (0.07)

Die Ergebnisse - Vermaschte Netze

→ Motivation
→ Netzwerkgraph
→ Schedule-Qualität
→ Planung
→ Ergebnisse

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	with prediction	without prediction	with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)		21 (0.64)
2	-	-	591 (0.32)		171 (0.29)
3	-	-	2096 (0.22)		351 (0.21)
					820 (0.13)
					1711 (0.09)
					3081 (0.07)



Die Ergebnisse - Vermaschte Netze

[→ Motivation](#)
[→ Netzwerkgraph](#)
[→ Schedule-Qualität](#)
[→ Planung](#)
[→ Ergebnisse](#)

Network type	Complete Search		Greedy Best-first search		
	all routes	best route	all routes with prediction	all routes without prediction	best route with or without prediction
1	276480 (0.63)	4320 (0.66)	42 (0.64)	21 (0.65)	21 (0.64)
2	-	-	591 (0.32)	612 (0.32)	171 (0.29)
3	-	-	2096 (0.22)	2200 (0.25)	351 (0.21)
4	-	-	-	-	820 (0.13)
5	-	-	-	-	1711 (0.09)
6	-	-	-	-	3081 (0.07)

Fazit & Ausblick

Fazit & Ausblick

Mit einfachen Suchheuristiken können bereits für kleinere Netze Planungen durchgeführt werden.

Die Suche mit nur der besten Route für eine Nachricht reduziert den Aufwand, ohne dass die Qualität der Schedules deutlich abnimmt.

Fazit & Ausblick

Mit einfachen Suchheuristiken können bereits für kleinere Netze Planungen durchgeführt werden.

Die Suche mit nur der besten Route für eine Nachricht reduziert den Aufwand, ohne dass die Qualität der Schedules deutlich abnimmt.



Weitere Suchheuristiken sollen untersucht werden

Fazit & Ausblick

Mit einfachen Suchheuristiken können bereits für kleinere Netze Planungen durchgeführt werden.

Die Suche mit nur der besten Route für eine Nachricht reduziert den Aufwand, ohne dass die Qualität der Schedules deutlich abnimmt.



Weitere Suchheuristiken sollen untersucht werden

Aufteilung des Netzwerks in (Linien-) Subnetzwerke, da für Linien der optimale Schedule berechnet werden kann