

Wahlalgorithmen

Motivation

Grundlagen

Wellenverfahren

Adoptionsverfahren



- Problem: Wahl eines Anführerknotens
 - Beispielszenarien
 - Koordinierung verteilter Aktionen
 - Erzeugung systemweit eindeutiger Token
 - Anlässe
 - Initialisierung im Rahmen des Systemstarts
 - Neukonfigurierung als Reaktion auf Fehler
 - Verwandtes Problem: Bestimmung einer maximalen Knoten-ID
- Anforderungen
 - *Eindeutigkeit*: Zu jedem Zeitpunkt ist maximal ein Knoten der Anführer
 - *Terminierung*: Bestimmung des Anführers erfolgt in endlicher Zeit
- Zusätzliche Kriterien (Beispiele)
 - Deterministischer Wahlalgorithmus
 - Benachrichtigung aller Knoten über das Ende bzw. Ergebnis der Wahl



■ Systemmodell

- Verteiltes System mit potenziell sehr vielen Knoten
 - Knoten
 - Gesamtzahl aller Knoten ist unbekannt
 - Jeder Knoten hat eindeutige ID
 - Nicht alle Knoten sind kontinuierlich Teil des Systems
 - Unterschiedliche Netztopologien
 - Ring
 - Baum
 - Beliebige Strukturierung
- Nicht jeder Knoten ist notwendigerweise mit jedem anderen verbunden

■ Herausforderungen

- Wie kann eine Wahl in nicht vollvermaschten Systemen realisiert werden?
- Wie lässt sich die Effizienz durch Wissen über die Netztopologie steigern?
- Wie wählt man einen Anführer in Systemen mit komplexer Netztopologie?



Erzeugung eines virtuellen Baums beim Systemstart


- Vorbereitung
 - Ermittlung einiger Adressen anderer Knoten für Verbindungsaufbau
 - Beispiel: Nutzung einer Registry
- Verbindungsaufbau
 - Verwaltung eines lokalen Levels
 - Wurzelknoten des Baums hat Level 0
 - Elternknoten: Nachbar mit kürzester Distanz zum Wurzelknoten
- Wiederholung der Auswahl bei Abbruch der Elternverbindung

```
private int level = -1;
private VSNode parent = null;

public void connect(List<VSNode> nodes) {
    for(VSNode node: nodes) {
        // Verbindungsaufbau
        boolean connected = [Aufbau der Verbindung];
        if(!connected) continue;

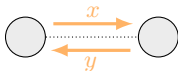
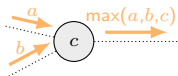
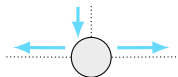
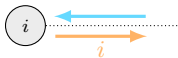
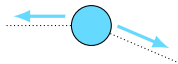
        // Bestimmung des Elternknotens
        if((level < 0) || (level > node.level)) {
            level = node.level + 1;
            parent = node;
        }
    }
}
```



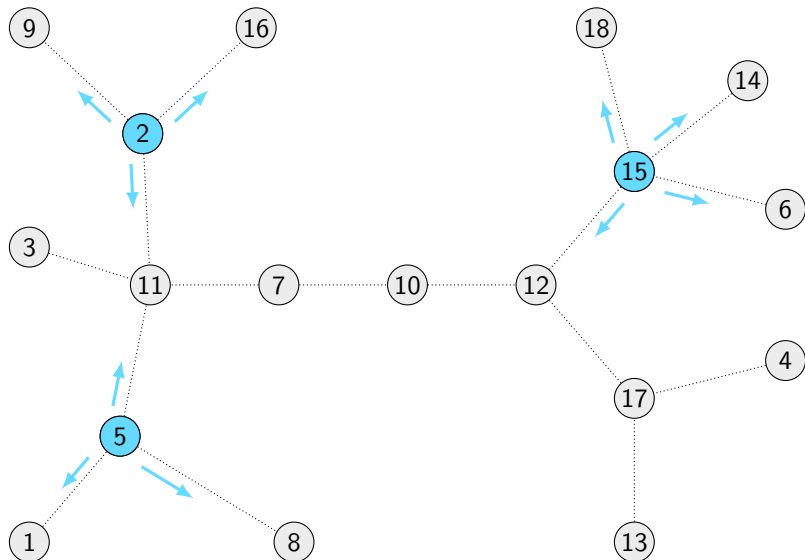
- Baumstruktur
 - Ungerichtete Kanten
 - Keine Zyklen
- Eigenschaften
 - Kommunikation mit unterschiedlich vielen Nachbarknoten
 - Potenzial für *Divide-and-Conquer*-Ansätze
- Wellenverfahren
 - *Explorationswelle* durchläuft den Baum bis zu den Blättern
 - *Echoweile* kommt zurück und bestimmt die höchste ID
 - *Informationswelle* teilt allen Knoten das Ergebnis mit
- Literatur
 -  Friedemann Mattern
Verteilte Basisalgorithmen
Springer-Verlag, 1989.



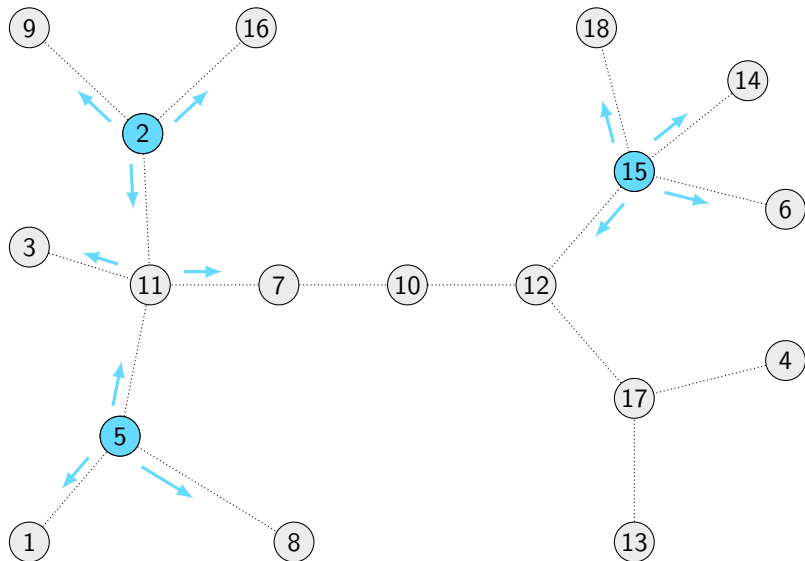
- Dedizierte Initiator-knoten senden *Explorernachrichten* an alle ihre Nachbarknoten
- Blattknoten senden bei Empfang einer Explorernachricht die eigene ID als Echonachricht zurück
- Innere Knoten mit k Kanten
 - Weiterleitung der ersten Explorernachricht in alle übrigen Richtungen
 - Nach Erhalt von Echonachrichten auf $k - 1$ Kanten: Senden einer Echonachricht e mit dem Maximum aller bisher bekannten IDs über die verbleibende Kante
 - Falls Empfang einer weiteren Echonachricht e' : Höchste ID im Netz ist Maximum der IDs von e und e'



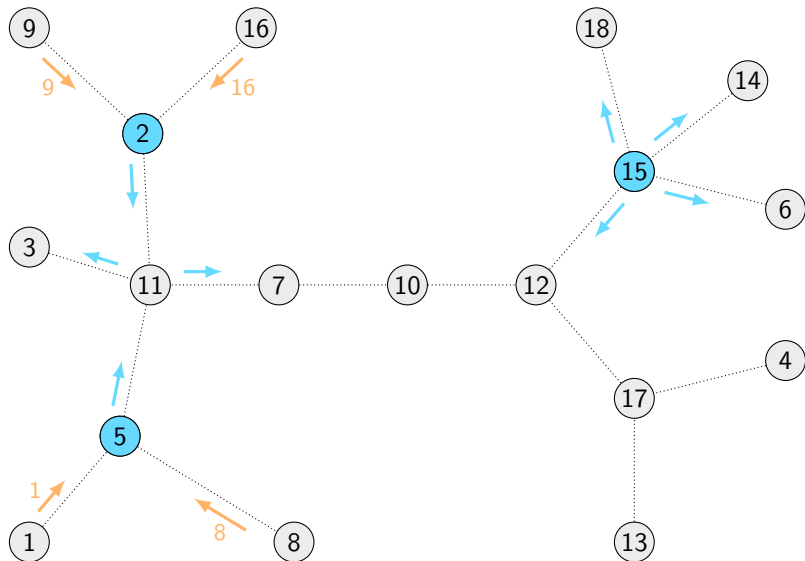
Beispiel



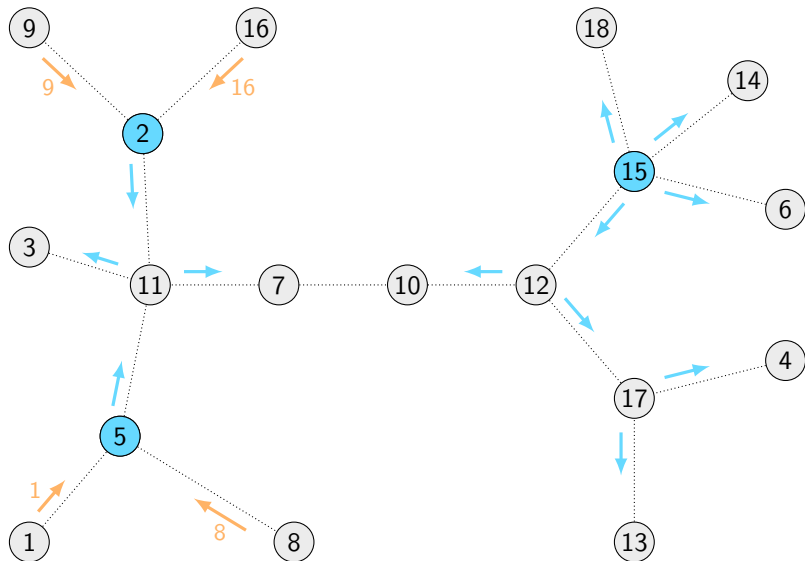
Beispiel



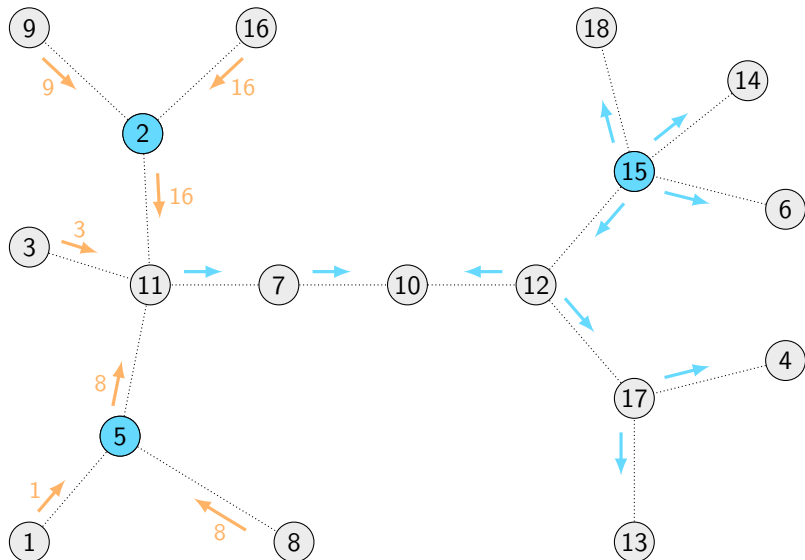
Beispiel



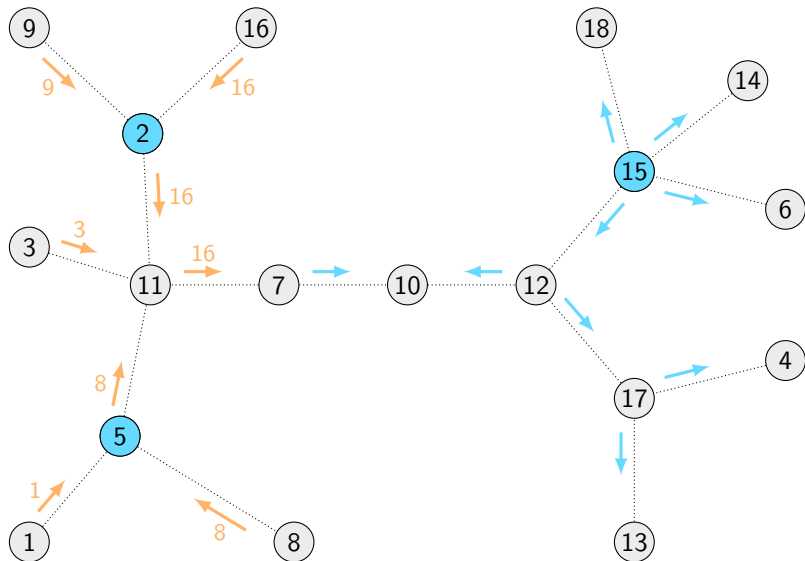
Beispiel



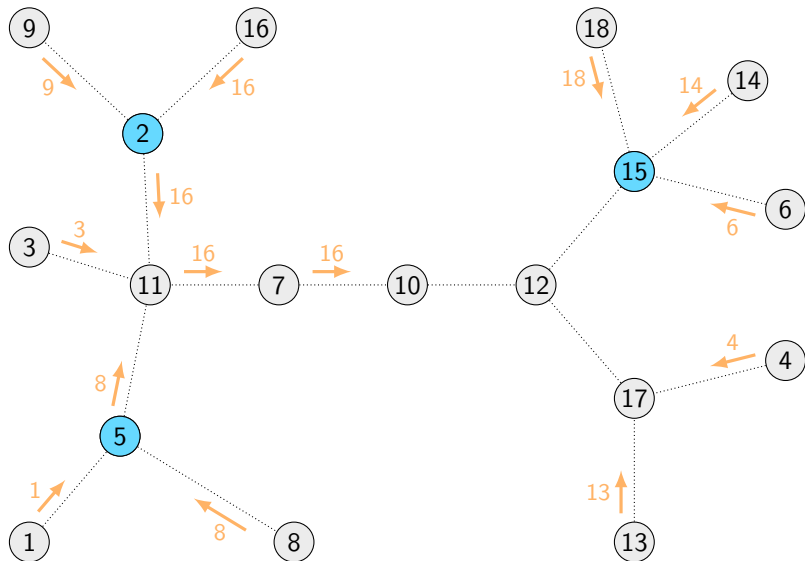
Beispiel



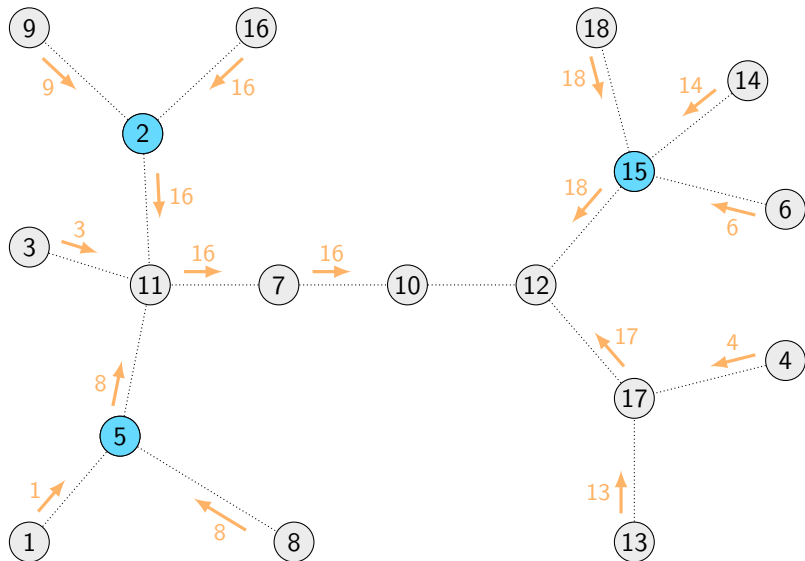
Beispiel



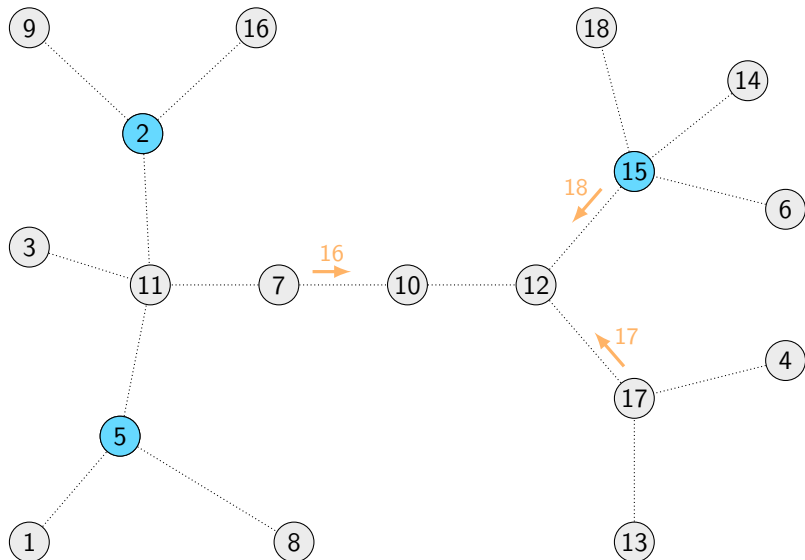
Beispiel



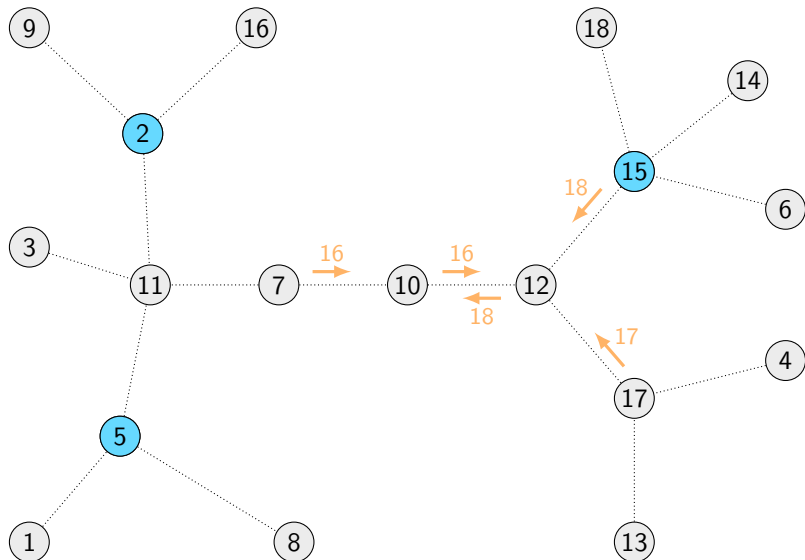
Beispiel



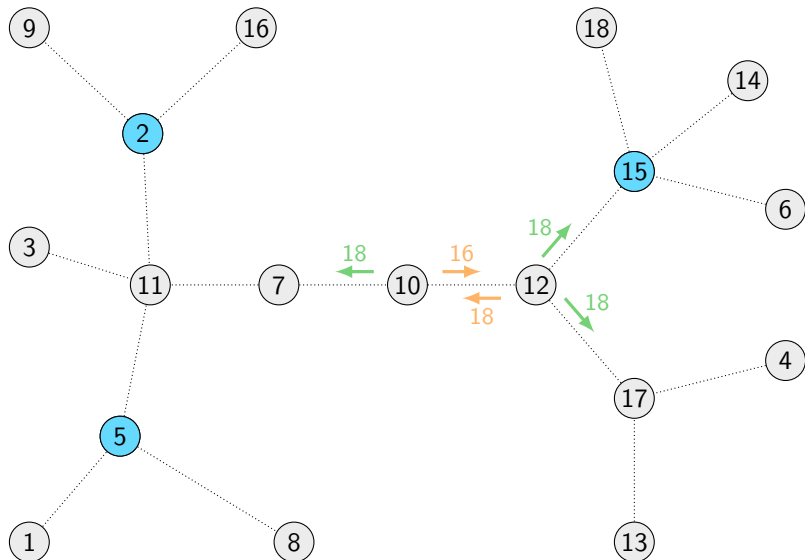
Beispiel



Beispiel



Beispiel

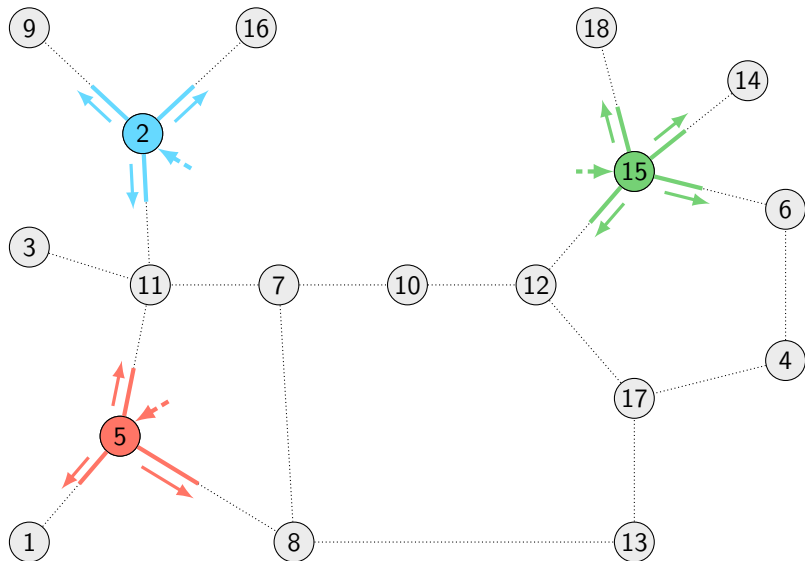


Adoptionsverfahren für beliebige Topologien

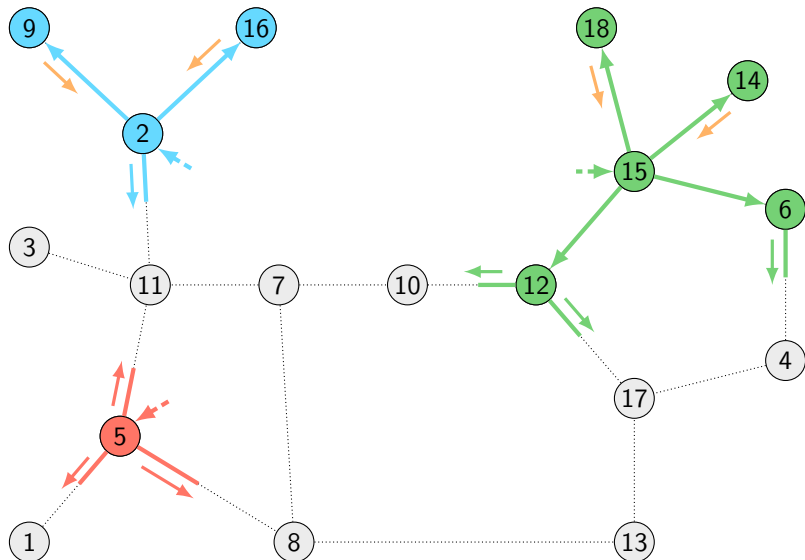
- Grundprinzip wie bei Wahl auf Bäumen: Explorations- und Echowelle
- Unterschiede
 - Explorer- und Echonachrichten tragen ID des zugehörigen Initiators
 - Nach Erhalt eines Echos: Markieren der Kante als erledigt („OK“)
- Verwaltung von IDs an Knoten und Kanten
 - Speicherung der höchsten einem Knoten bekannten Initiator-ID (ID_i)
 - *Elternkante*: Kante, über die Explorernachricht mit ID_i empfangen wurde
 - Markieren der Sendekante einer Explorernachricht mit Initiator-ID
- Empfang einer Explorernachricht x über Kante k mit Markierung m
 - Falls $ID_x = ID_m$ Kante k ist nicht Teil des virtuellen Baums
 - Falls $ID_x < ID_m$ Ignorieren der eintreffenden Explorernachricht x
 - Falls $ID_m < ID_x \leq ID_i$ Aktualisierung von m , Senden von x über k
Falls $ID_x = ID_i$: k ist nicht Teil des Baums
 - Falls $ID_m \leq ID_i < ID_x$ *Adoption*: k wird neue Elternkante, $ID_i := ID_x$
Weiterleitung von x über bisherige Elternkante



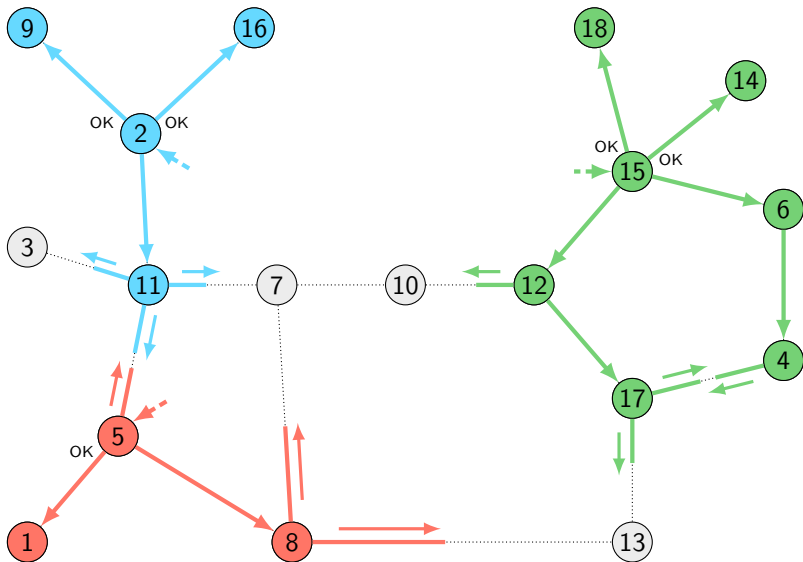
Beispiel



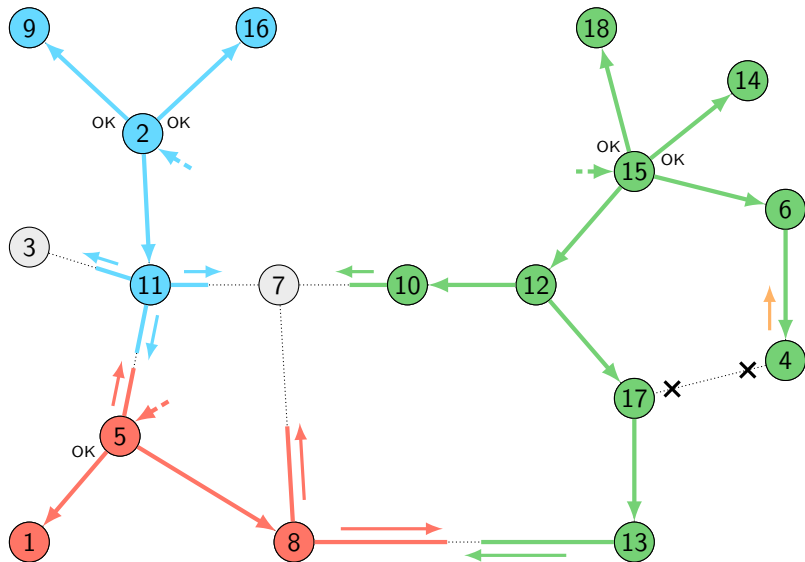
Beispiel



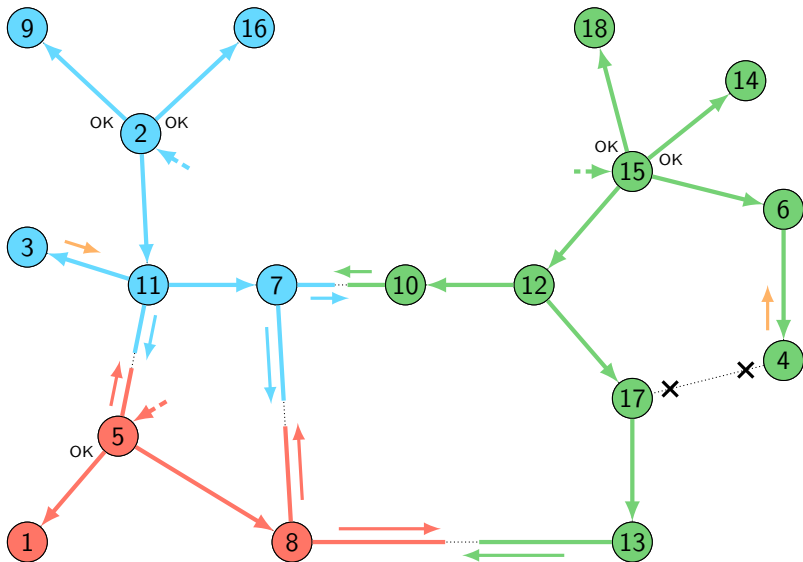
Beispiel



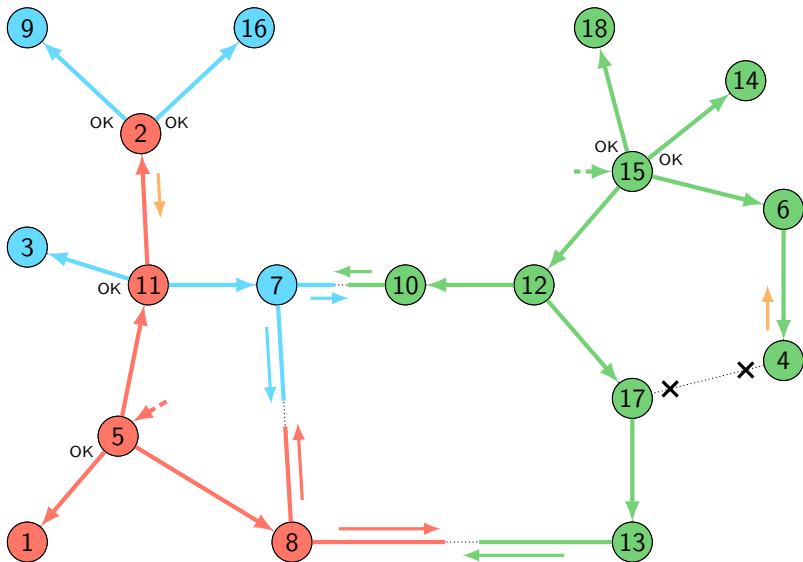
Beispiel



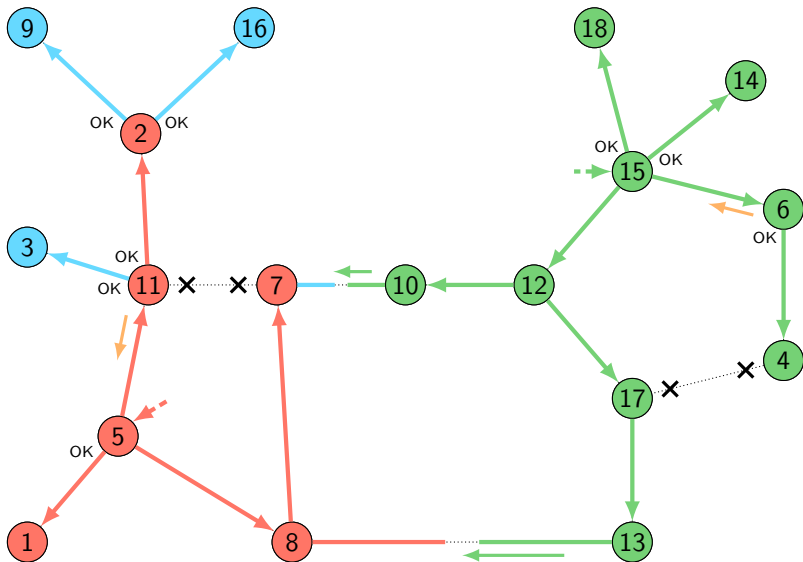
Beispiel



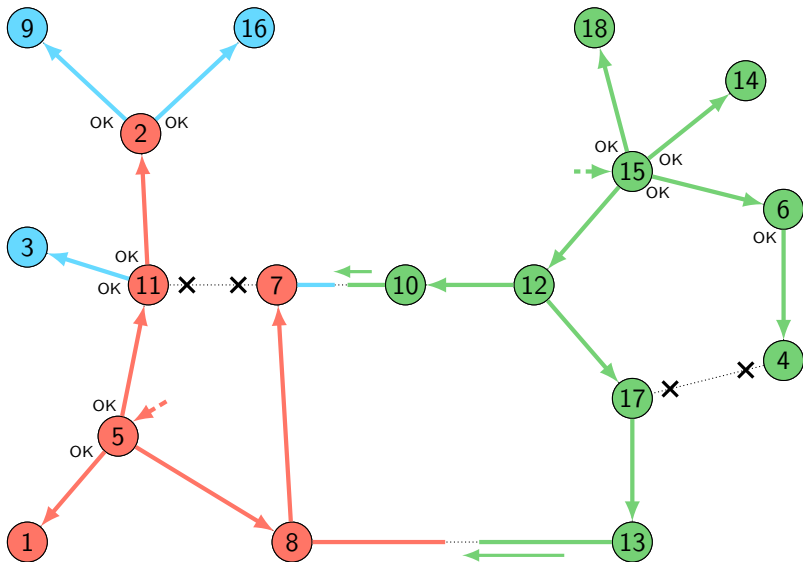
Beispiel



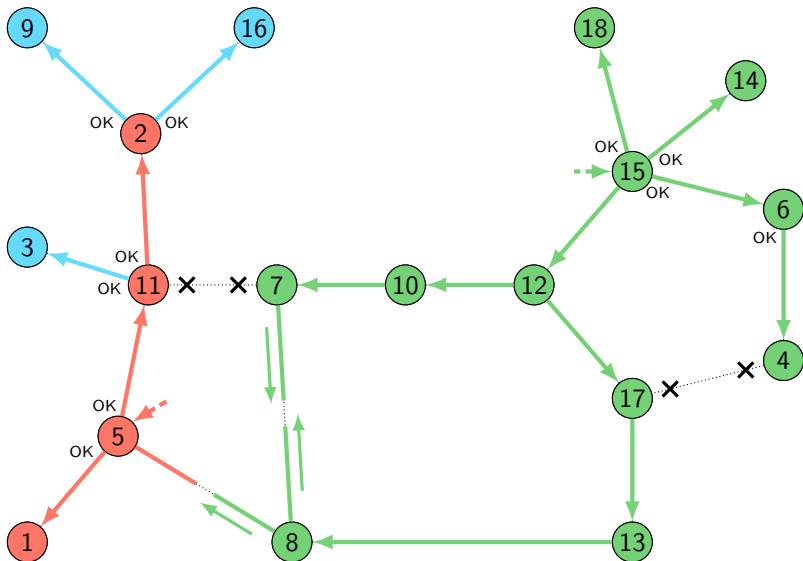
Beispiel



Beispiel



Beispiel



Beispiel

