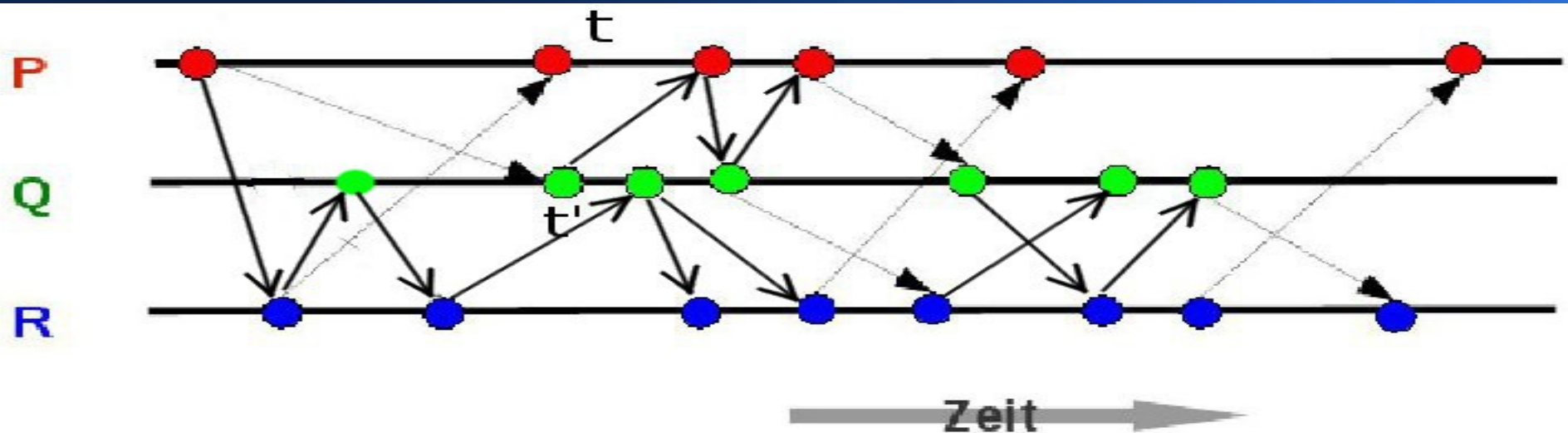


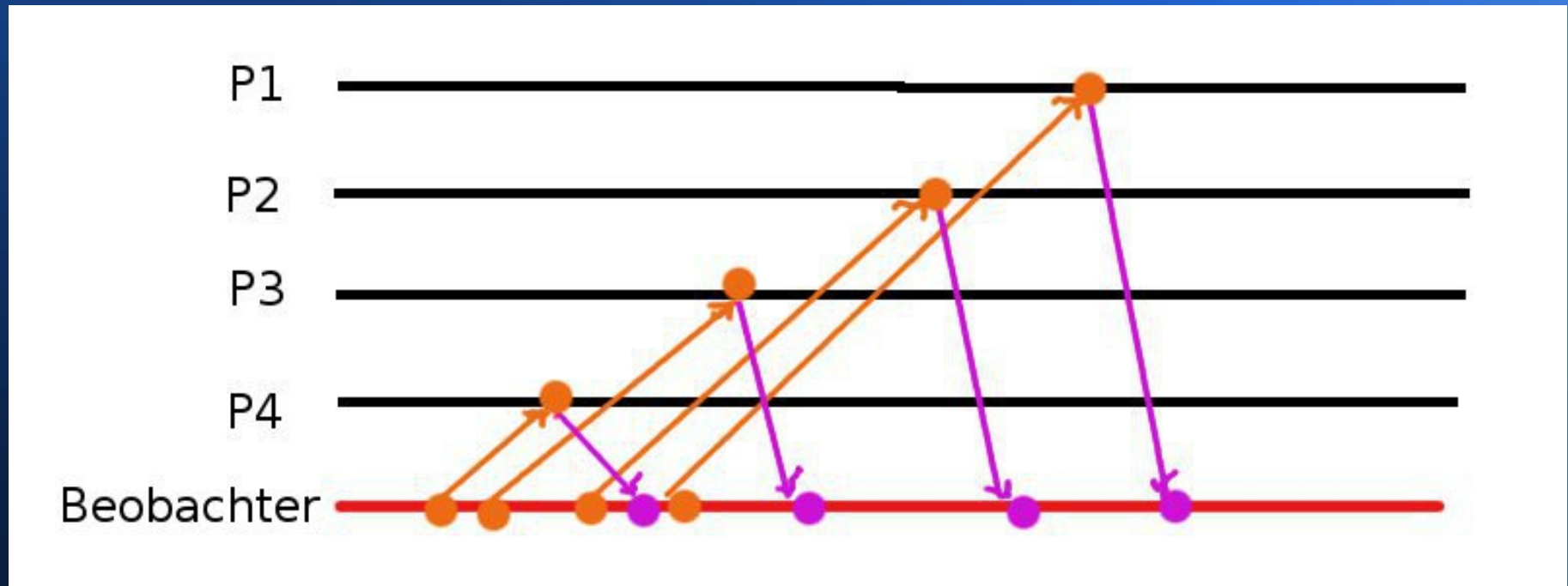
Terminierung

Atommodell:

- Prozess sendet nicht spontan
- Prozess reagiert auf Nachrichten
- Initiator nötig

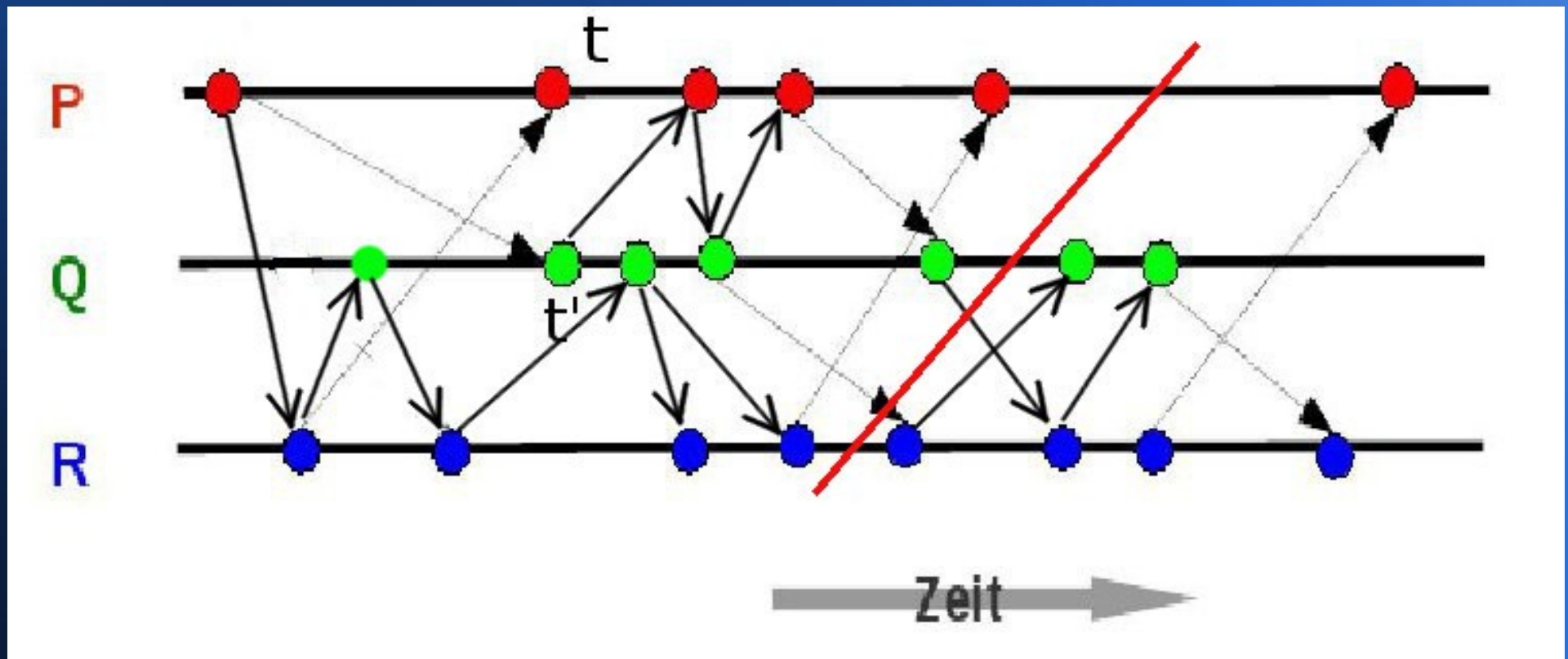


Modell für Terminierungsalgorithmen



Beobachterprozess, der jeden Knoten kontaktieren kann

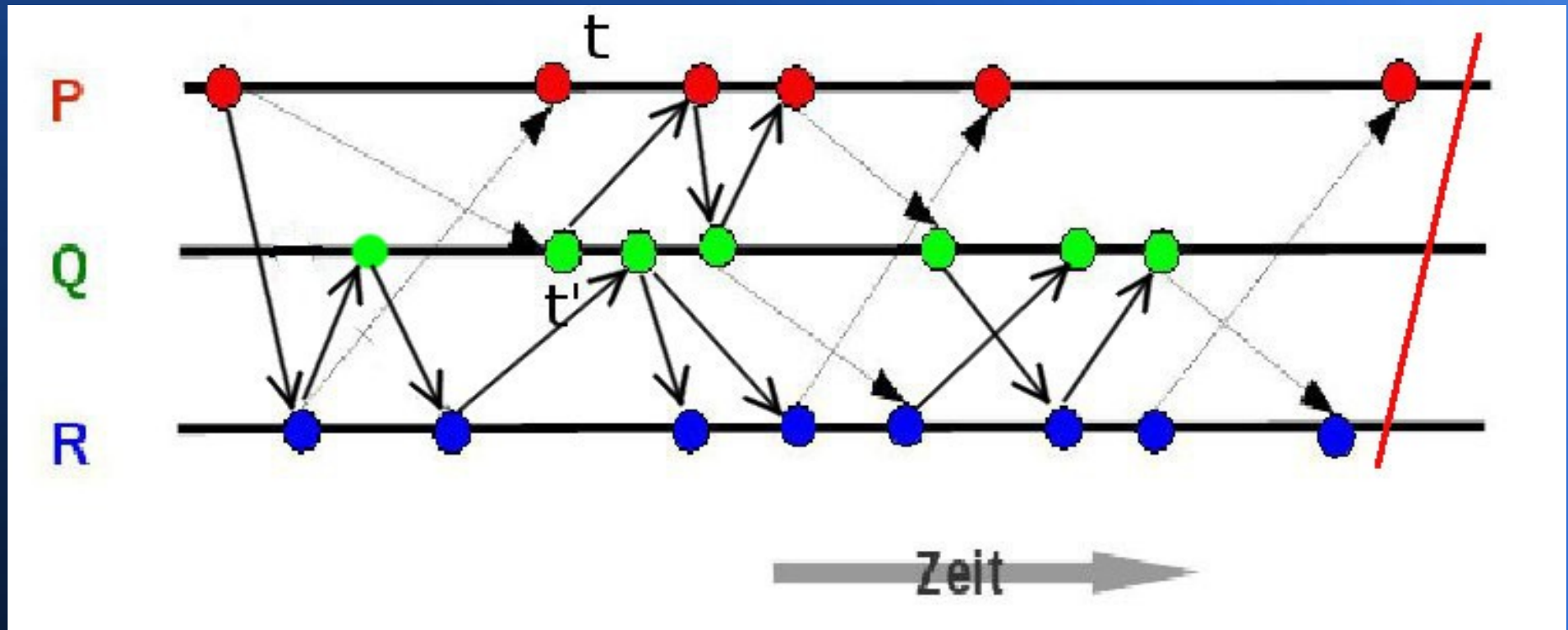
Terminierung: einfache Zählung



Algorithmus noch aktiv:

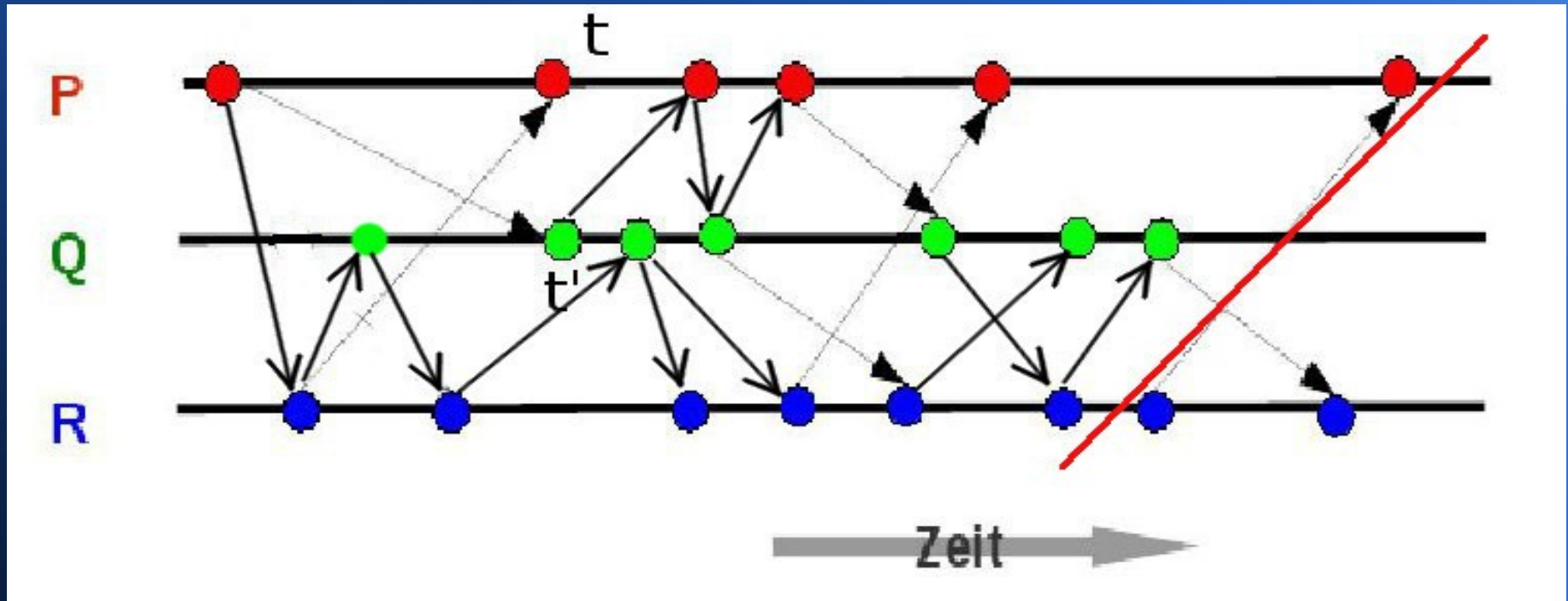
insgesamt mehr gesendete als empfangene Nachrichten

Terminierung: einfache Zählung



Gesamtanzahl gesendeter = Gesamtanzahl empfangene Nachrichten

Terminierung: einfache Zählung reicht nicht

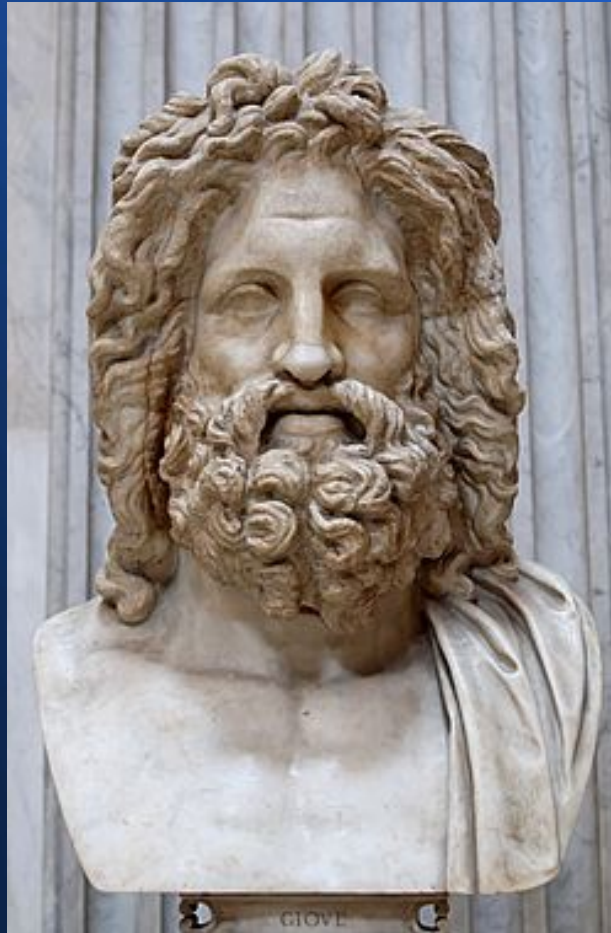


Insgesamt Anzahl gesendeter Nachrichten = Anzahl empfangener Nachrichten

$$S = R$$

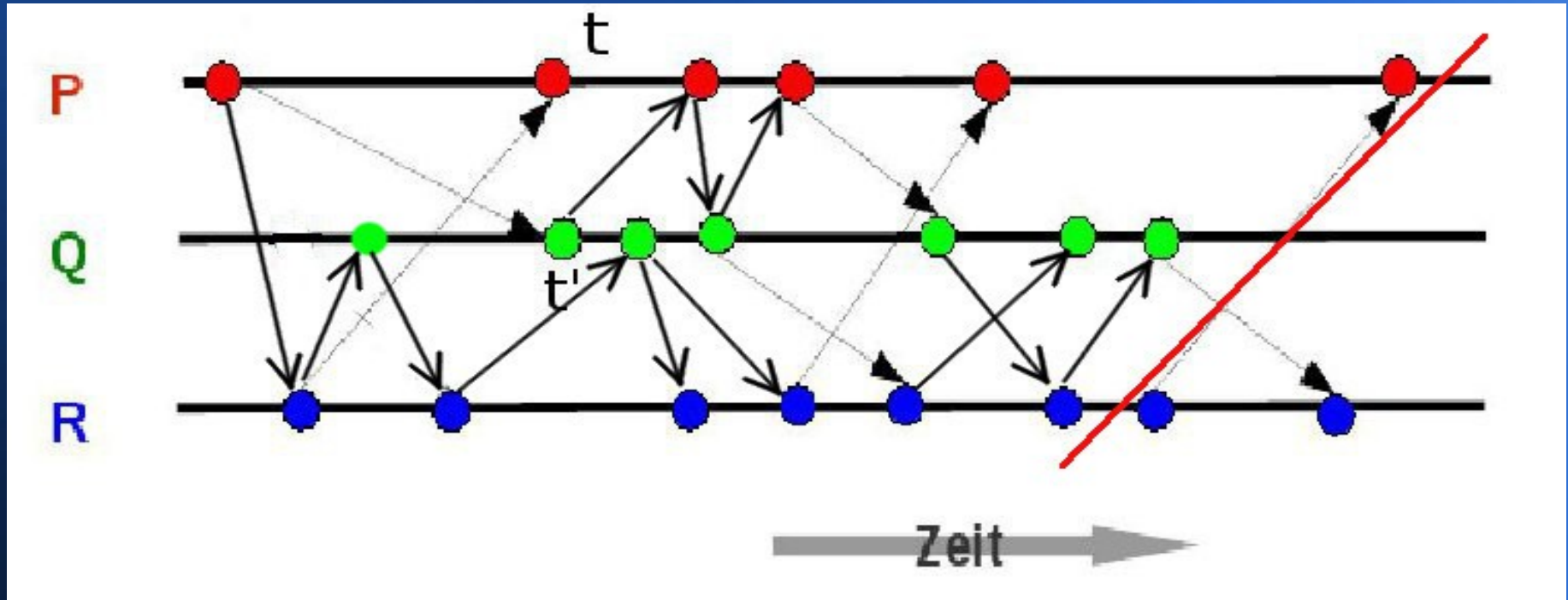
Was tun?

sprach Zeus ...



(aus: "Die Teilung der Erde" von Friedrich Schiller)

Genauer zählen!

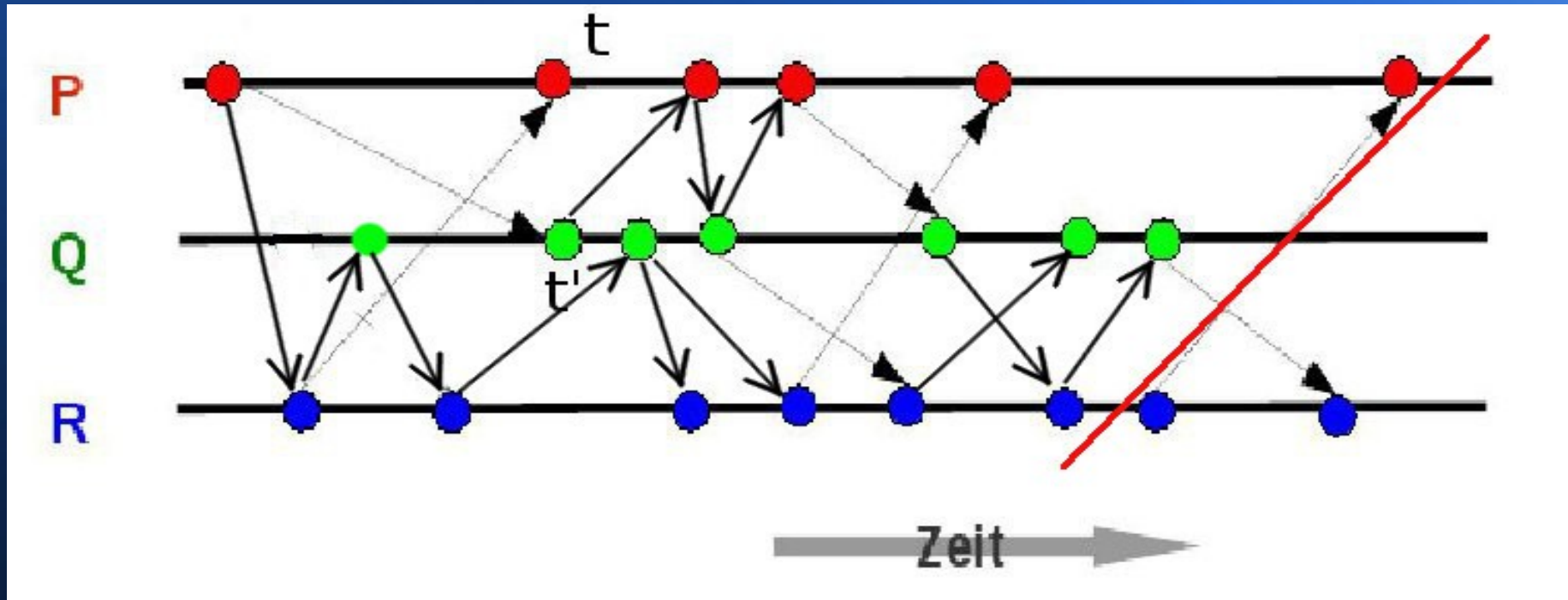


Beobachtung 1:
eine von R nach P gesendete Nachricht
empfangen, aber noch nicht als gesendet
gezählt

Beobachtung 2:
eine von Q nach R gesendete Nachricht
gezählt, aber noch nicht als empfangen
gezählt

Speicherplatzverbrauch?

Kanalzähler

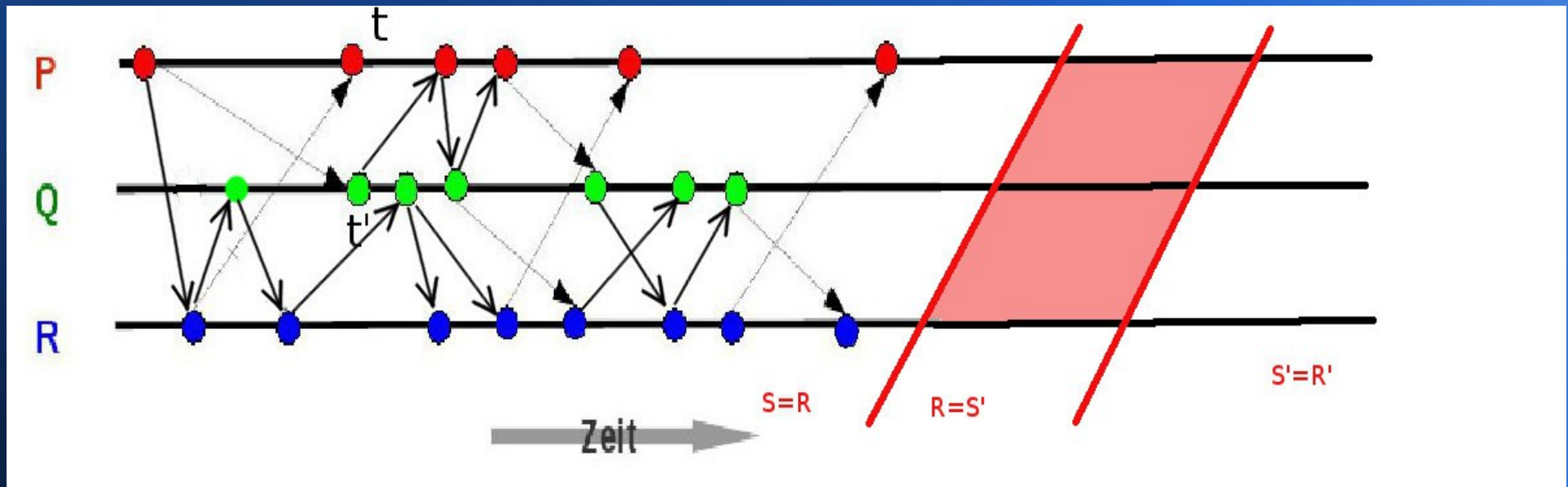


Beobachtung 1:
eine von R nach P gesendete Nachricht
empfangen, aber noch nicht als gesendet
gezählt

Beobachtung 2:
eine von Q nach R gesendete Nachricht
gezählt, aber noch nicht als empfangen
gezählt

n Prozesse jeweils $2n$ Zähler

Doppelzählung



$$S=R=S'=R'$$

Doppelzählverfahren: Korrektheit

- **safety**: es passiert nichts Falsches
- zwischen den beiden Schnitten:
nichts gesendet, nichts empfangen
- es kann also höchstens Nachrichten
 - komplett vor dem Schnitt oder
 - komplett nach dem Schnitt oder
 - beide Schnitte überquerend geben
- **liveness**: es passiert schließlich das Richtige
zum Terminierungszeitpunkt gilt $S=R=S'=R'$

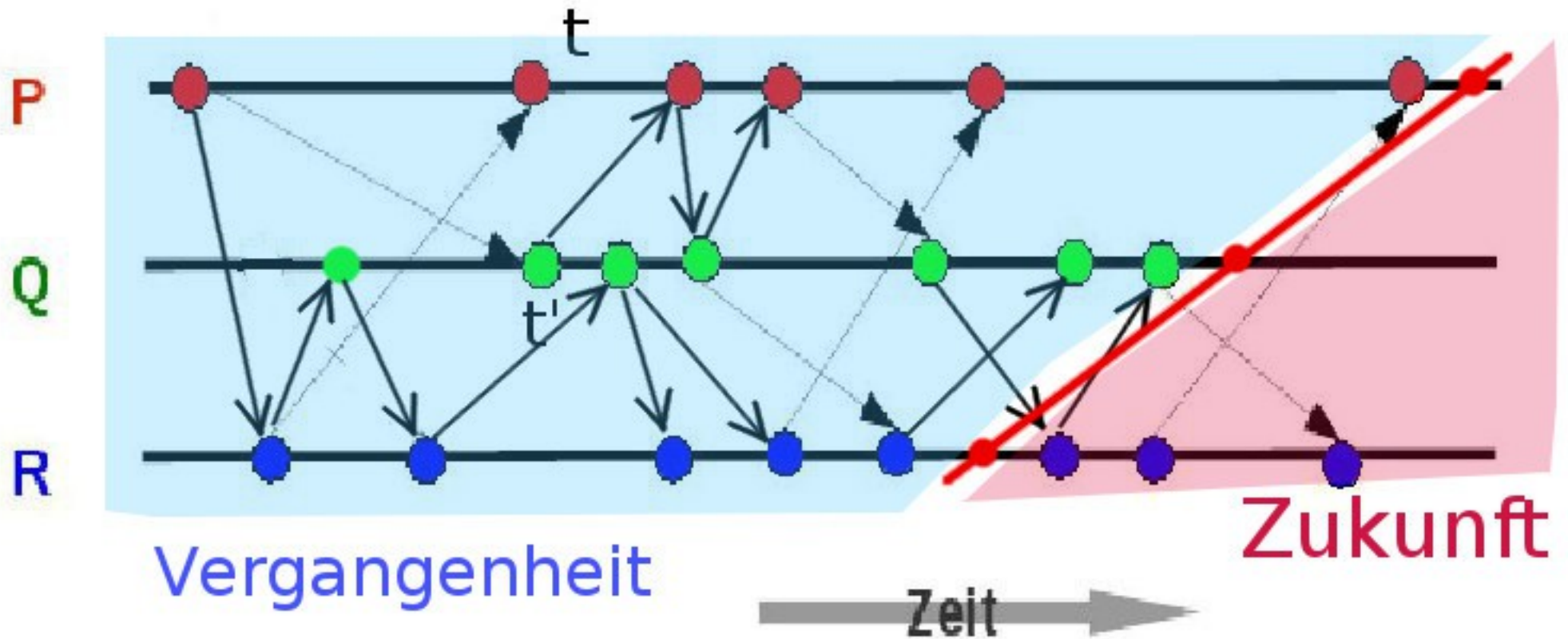
Zeitzoneverfahren: trennt Vergangenheit und Zukunft



Zurück in
die
Zukunft !!!

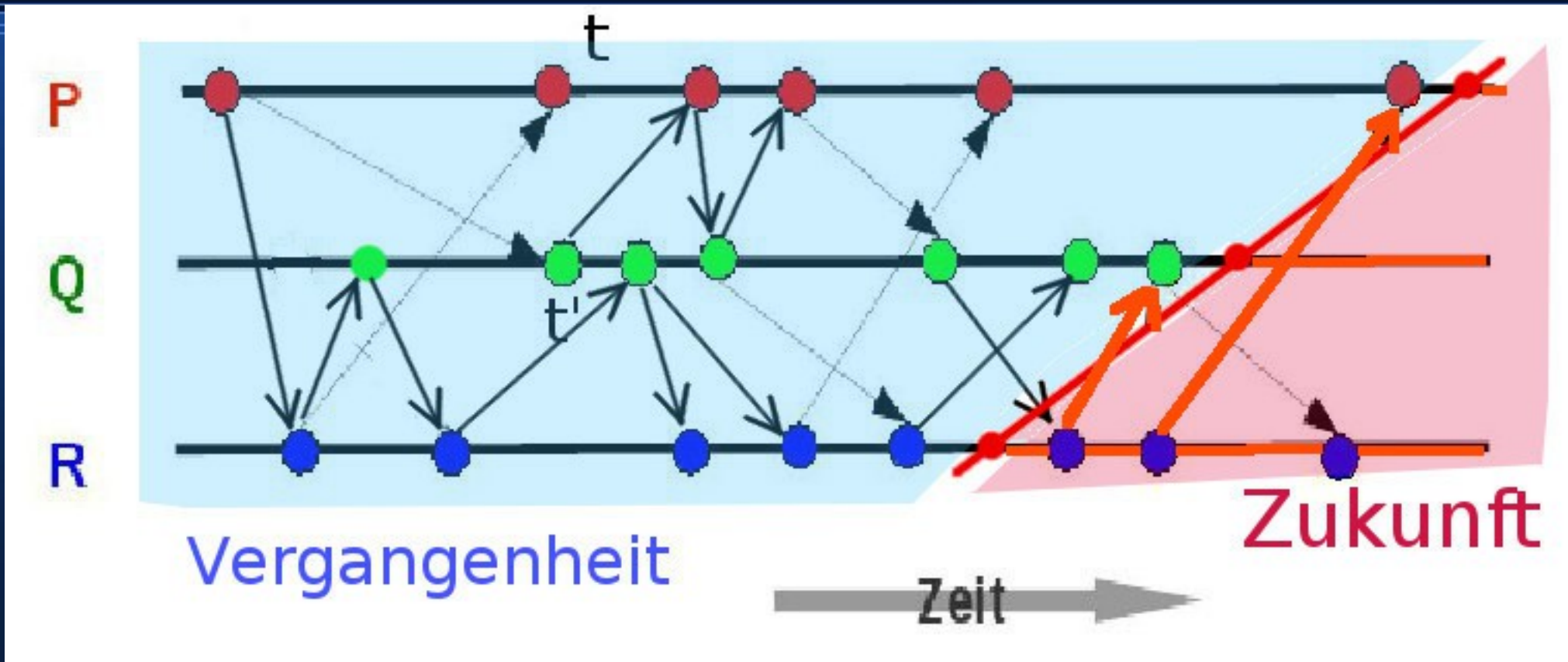
Der
Fluxkompensator
!!!

Zeitzoneverfahren



Prozess R in roter Zeitzone, sobald Beobachternachricht eingetroffen

Zeitzoneverfahren



R färbt von da an seine Nachrichten rot

Rote Nachricht in blauer Zeitzone eingetroffen

Zählung ungültig, wiederhole die Zählung

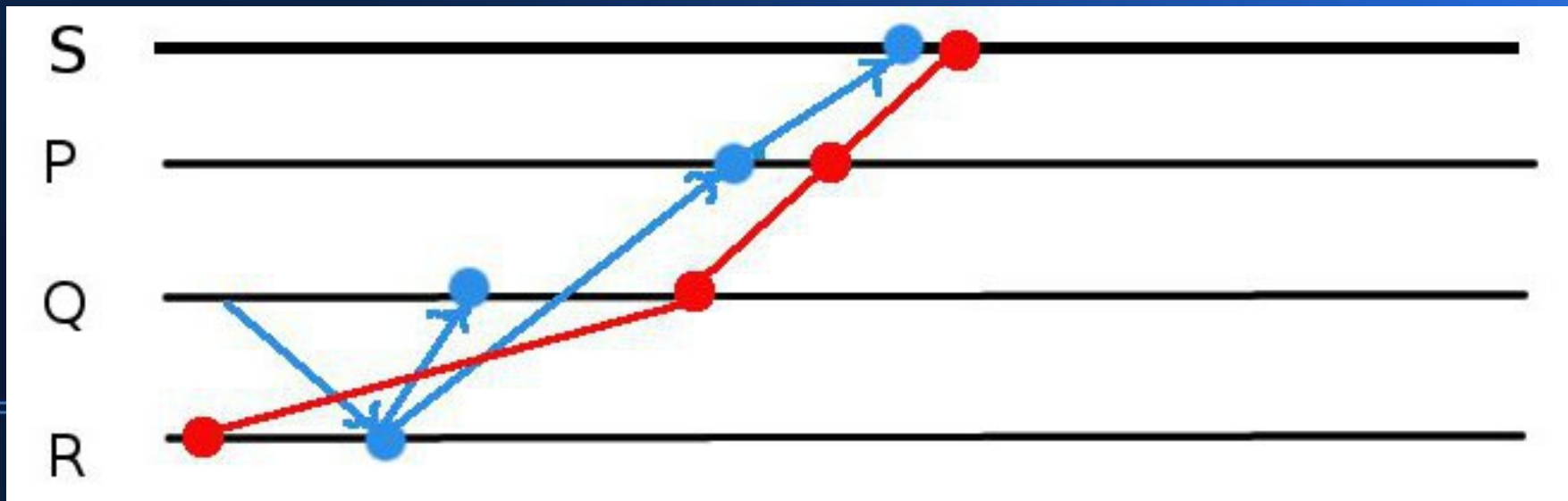


Zeitzoneverfahren

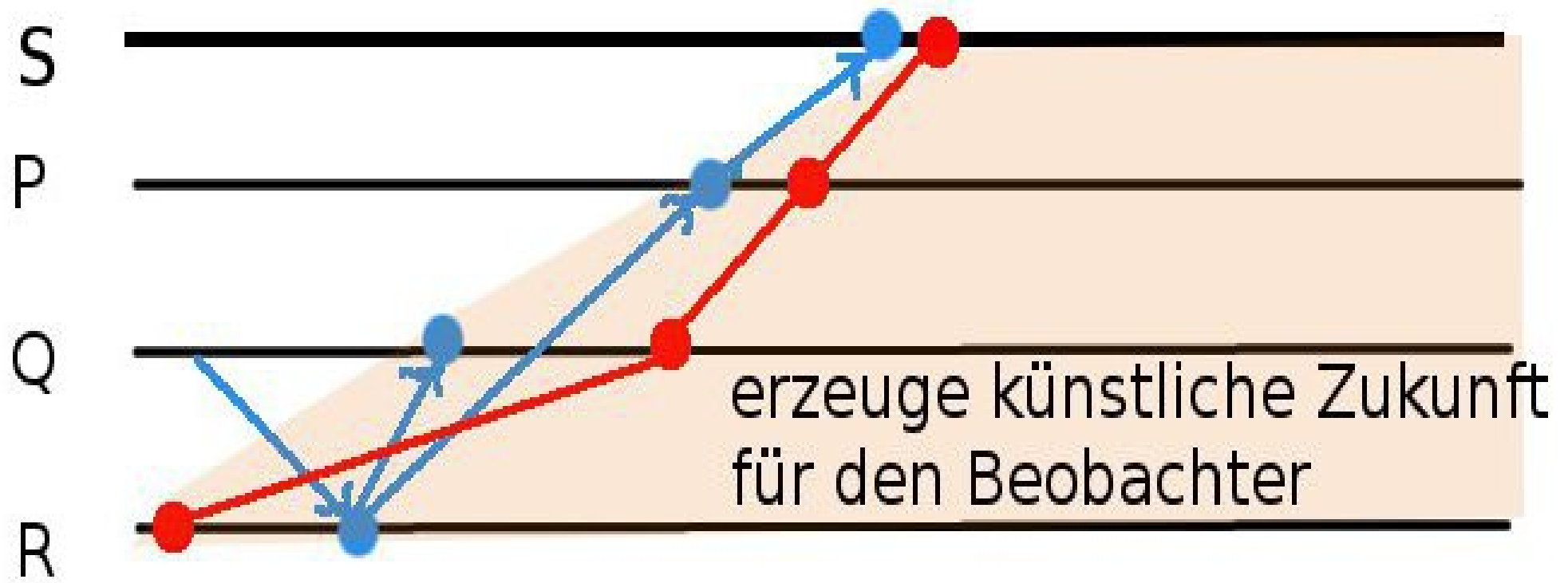
- falls schon terminiert, nur einmal zählen
- Nachrichten markieren, ob sie
 - in der Vergangenheit oder
 - in der Zukunftabgesendet wurden
- Verallgemeinerung: Knoten in Zeitzone z
- Nachricht vom Beobachter trifft ein
 - Knoten nun in Zeitzone $z+1$

Zeitzoneverfahren mit Trick: vermeide ungültige Zählungen

- wenn Nachricht aus der Zukunft erkannt
 - Rette den Zählerstand
 - Beobachter später, erhält geretteten Stand
 - gesendete Zukunftsnachrichten bleiben in Zukunft



Zeitzoneverfahren: der Trick



Vektorverfahren

Vektoren, ähnlich der Vektorzeit

hier Nachrichtenzähler

Beobachtung: Kanalzähler zählen zuviel

Fasse Empfangszähler zusammen

(es ist nicht wichtig, von wem man empfängt)

Vektorverfahren

Vektor für Prozess Nr. i

gesendete

Hier empfangene NR zählen

i -te Komponente summiert gesendete an i

außer beim Empfänger: Summe negativ

$$\begin{pmatrix} S_1 \\ S_2 \\ S_3 \\ \vdots \\ S_{i-1} \\ -R \\ S_{i+1} \\ \vdots \\ S_n \end{pmatrix}$$

Vektorverfahren Implementierung

- Initialisiere *summenvektor* = 0
- für jeden Knoten i sende Abfrage
 - erhalte Vektor $v(i)$
 - Addiere *summenvektor* = *summenvektor* + $v(i)$
- falls *summenvektor* = 0,
 - dann Berechnung terminiert

Zusammenfassung

Verfahren	Anzahl Zählungen	Anzahl Zähler
Kanalzähler	1	$2n^2$
Doppelzählverfahren	2	4
Zeitzonenverfahren	1	2 (+Zeitzonenflags)
Vektorverfahren	1	n^2