

4. Aufgabenblatt: Echo-Algorithmus

- Filesharing-Dienst mit Peer-to-Peer-Topologie
- Station wird gestartet und verbindet sich mit Nachbarn
- Benutzer auf bel. Station stellt Suchanfrage
- Probe-Echo-Algorithmus durchsucht gesamtes (unbekanntes!) Netzwerk und liefert Ergebnisliste
- Benutzer kann ausgewählte Datei herunterladen

```

monitor Station(Set<Station> neighbours)
    boolean engaged    := false;
    Station pred       := nil;
    int    replies     := 0;

proc broadcast()
    engaged := true;
    foreach n in neighbours do n.probe(this)
    await replies = |neighbours|;

port probe(Station s)
    if !engaged then // bei Zyklen probe als echo werten
        engaged := true; pred := s;
        foreach n in neighbours-pred do n.probe(this);
    replies := replies + 1;
    if replies = |neighbours| then
        engaged := false;
        if pred != nil then pred.echo(); // kein pred bei
                                           // initiator

port echo()
    replies := replies + 1;
    if replies = |neighbours| then
        engaged := false;
        if pred != nil then pred.echo();

```

Schnappschuss-Algorithmen

- Eine Instanz möchte ein Abbild des gesamten verteilten Systems herstellen.
- Zustand kann nur durch Nachrichtenaustausch erfragt werden
- Ideal: überall gleichzeitig erfasst
- Hoher Aufwand, daher evtl. auch nicht-gleichzeitige/veraltete Schnappschüsse akzeptabel
- Schnappschuss sollte jedoch *konsistent* sein
(existiert *haben können*)

Naiver Algorithmus

- **Beispiel: verteilter Algorithmus mit 3 Stationen. Es gibt eine „Population“ von 12 Einheiten, die durch Nachrichtenaustausch hin- und herwandern**
- **Initiator schickt Schnappschuss-Aufforderung an alle Stationen und sammelt die Antworten ein**
- **Je nach Laufzeit der Nachrichten kann Summe der Populationen auf den erfassten Abbildern der Stationen 12 übersteigen
-> Zustand kann so zu keinem Zeitpunkt existiert haben**

Schnappschuss unter Einfrieren

- Station wird nach Abschicken ihres Abbildes eingefroren
- Nachrichten stauen sich
- Initiator schickt Abschlussnachricht -> Auftauen

- Liefert idealen Schnappschuss
 - Zustand könnte existiert haben, Nachrichten „waren gerade unterwegs“
 - Zustand ist aktuell (weil Stationen eingefroren)
- sehr teuer

3. Variante

- Initiator schickt Aufforderung
- Nachdem Station Abbild abgeschickt hat, versieht sie alle Nachrichten des verteilten Algorithmus mit einer Markierung „erzeuge Schnappschuss“
- Empfänger erzeugt Schnappschuss noch bevor er die eigentliche Nachricht verarbeitet (durch diese droht Inkonsistenz, Verarbeitung jetzt wie der Versand nach dem Schnappschuss)
- Echte Schnappschuss-Aufforderung wird ggf. ignoriert
- Schnappschuss nicht notwendigerweise aktuell, aber konsistent