Let there be $h=6$ hosts (lockss-0, lockss-1, lockss-2, props, dris, haar), and $n=4$ AU’s (AU0, AU1, AU2, AU3).

Generate $h$ (6) invitations, such that:

(A) For each AU, there are at least $k=4$ hosts harvesting

(B) For each host, the sum of max size of AU’s harvested is less than storage commitment.

For AU0 (size=95):
* Invite lockss-0 ($k=1$) (harvesting 95, comm 100)
* Invite lockss-1 ($k=2$) (harvesting 95, comm 1700)
* Invite lockss-2 ($k=3$) (harvesting 95, comm 1700)
* Cannot invite props (harvesting 0, comm 39)
* Cannot invite dris (harvesting 0, comm 39)
* Invite haar ($k=4$) (harvesting 95, comm 200)

For AU1 (size=18):
* Cannot invite lockss-0 (harvesting 95, comm 100)
* Invite lockss-1 ($k=1$) (harvesting 113, comm 1700)
* Invite lockss-2 ($k=2$) (harvesting 113, comm 1700)
* Invite props ($k=3$) (harvesting 18, comm 39)
* Invite dris ($k=4$) (harvesting 18, comm 39)

For AU2 (size=18):
* Cannot invite lockss-0 (harvesting 95, comm 100)
* Invite lockss-1 ($k=1$) (harvesting 131, comm 1700)
* Invite lockss-2 ($k=2$) (harvesting 131, comm 1700)
* Invite props ($k=3$) (harvesting 36, comm 39)
* Invite dris ($k=4$) (harvesting 36, comm 39)

For AU3 (size=4):
* Cannot invite lockss-0 (harvesting 99, comm 100)
* Invite lockss-1 ($k=1$) (harvesting 135, comm 1700)
* Invite lockss-2 ($k=2$) (harvesting 135, comm 1700)
* Cannot invite props (harvesting 36, comm 39)
* Cannot invite dris (harvesting 36, comm 39)
* Invite haar ($k=4$) (harvesting 99, comm 200)
1. Update schema instance
2. Process updated instance
3. Send invitations
4. Accept invitations
5. Harvest AU's

UC1 – All invitations accepted
Flow Diagram
Data-PASS SSP Use Case 1 – Some Invitations Rejected

### SCHEME

Let there be $h=6$ hosts (lockss-0, lockss-1, lockss-2, props, dris, haar), and $n=4$ AU's (AU0, AU1, AU2, AU3).

Generate $h$ (6) invitations, such that:

- **(A)** For each AU, there are at least $k=4$ hosts harvesting
- **(B)** For each host, the sum of max size of AU's harvested is less than storage commitment.

For AU0 (size=95):
- Invite lockss-0 ($k=1$) (harvesting 95, comm 100)
- Invite lockss-1 ($k=2$) (harvesting 95, comm 1700)
- Invite lockss-2 ($k=3$) (harvesting 95, comm 1700)
- *Cannot invite props (harvesting 0, comm 39)
- *Cannot invite dris (harvesting 0, comm 39)
- *Invite haar ($k=4$) (harvesting 95, comm 200)

For AU1 (size=18):
- *Cannot invite lockss-0 (harvesting 95, comm 100)
- *Invite lockss-1 ($k=1$) (harvesting 113, comm 1700)
- *Invite lockss-2 ($k=2$) (harvesting 113, comm 1700)
- *Invite props ($k=3$) (harvesting 18, comm 39)
- *Invite dris ($k=4$) (harvesting 18, comm 39)

For AU2 (size=18):
- *Cannot invite lockss-0 (harvesting 95, comm 100)
- *Invite lockss-1 ($k=1$) (harvesting 131, comm 1700)
- *Invite lockss-2 ($k=2$) (harvesting 131, comm 1700)
- *Invite props ($k=3$) (harvesting 36, comm 39)
- *Invite dris ($k=4$) (harvesting 36, comm 39)

For AU3 (size=4):
- *Invite lockss-0 ($k=1$) (harvesting 99, comm 100)
- *Invite lockss-1 ($k=2$) (harvesting 135, comm 1700)
- *Invite lockss-2 ($k=3$) (harvesting 135, comm 1700)
- *Cannot invite props (harvesting 36, comm 39)
- *Cannot invite dris (harvesting 36, comm 39)
- *Invite haar ($k=4$) (harvesting 99, comm 200)

**k-values**
- AU0: 4
- AU1: 3
- AU2: 3
- AU3: 4

**Inadequate replication!**

Redetermine invitations:
- AU1 [size=18] [lockss-1, lockss-2, props]
- *Cannot invite lockss-0 (comm=100, harv=99)
- *Cannot invite dris (rejected invitation)
- *Can invite haar (comm=200, harv=99), $k=4$

- AU2 [size=18] [lockss-1, lockss-2, props]
- *Cannot invite lockss-0 (comm=100, harv=99)
- *Cannot invite dris (rejected invitation)
- *Can invite haar (comm=200, harv=117), $k=4$

Send new invitations:
- haar (AU0, AU1, AU2, AU3)
- haar accepts AU0, AU1, AU2, AU3

Invitations are sent to hosts:
- lockss-0 accepts AU0, AU3
- lockss-1 accepts AU0, AU1, AU2, AU3
- lockss-2 accepts AU0, AU1, AU2, AU3
- props accepts AU1, AU2
- *dris rejects AU1 and AU2*
  - haar accepts AU0, AU3


6/5/2008