Data-PASS SSP Use Case 2

Let there be \( h=6 \) hosts and \( n=4 \) AU's as previously defined in UC1.

Let there be \( j=3 \) new AU's (AU4, AU5, AU6) such that there are \( n+j=7 \) total AU's.

Generate \( h (6) \) new invitations, such that

**Z. The original \( n \) AU's assigned remain unchanged.**

(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.

Repopulate AU's in invitation from original invitation:

*AU0: Invite lockss-0, lockss-1, lockss-2, haar
*AU1: Invite lockss-1, lockss-2, props, dris
*AU2: Invite lockss-1, lockss-2, props, dris
*AU3: Invite lockss-0, lockss-1, lockss-2, haar

For AU5 (size=2): 
*Cannot invite lockss-0 (harvesting 100, comm 100)
*Invite lockss-1 (\( k=1 \)) (harvesting 138, comm 1700)
*Invite lockss-2 (\( k=2 \)) (harvesting 138, comm 1700)
*Invite props (\( k=3 \)) (harvesting 39, comm 39)
*Invite dris (\( k=4 \)) (harvesting 38, comm 39)

For AU6 (size=0.5):
*Cannot invite lockss-0 (harvesting 100, comm 100)
*Invite lockss-1 (\( k=1 \)) (harvesting 138.5, comm 1700)
*Invite lockss-2 (\( k=2 \)) (harvesting 138.5, comm 1700)
*Cannot invite props (harvesting 39, comm 39)
*Invite dris (\( k=3 \)) (harvesting 38.5, comm 39)
*Invite harr (\( k=4 \)) (harvesting 101.5, comm 200)