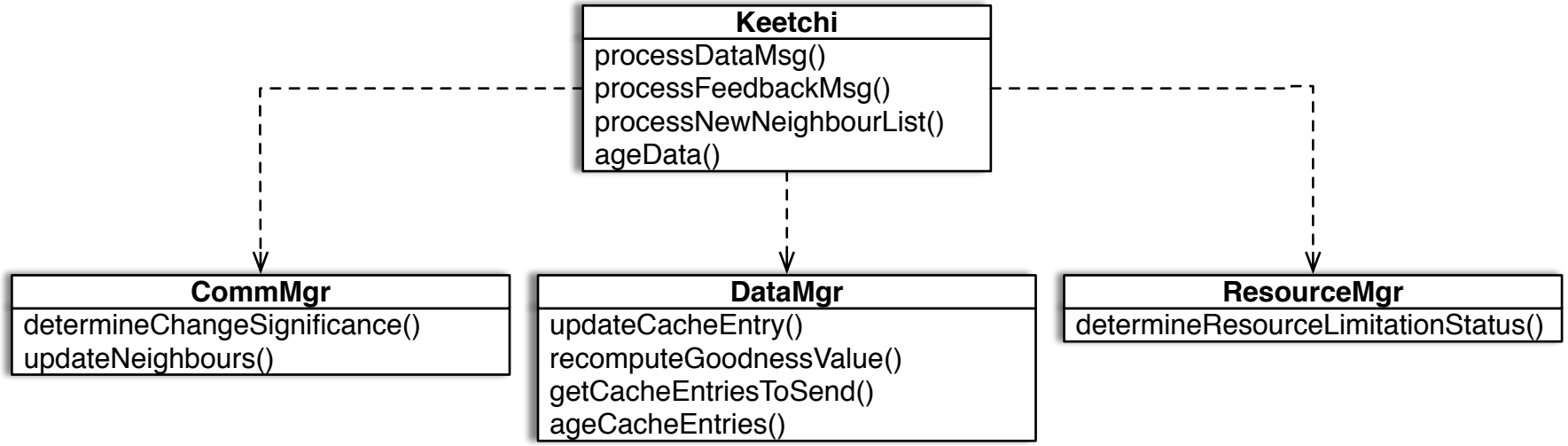
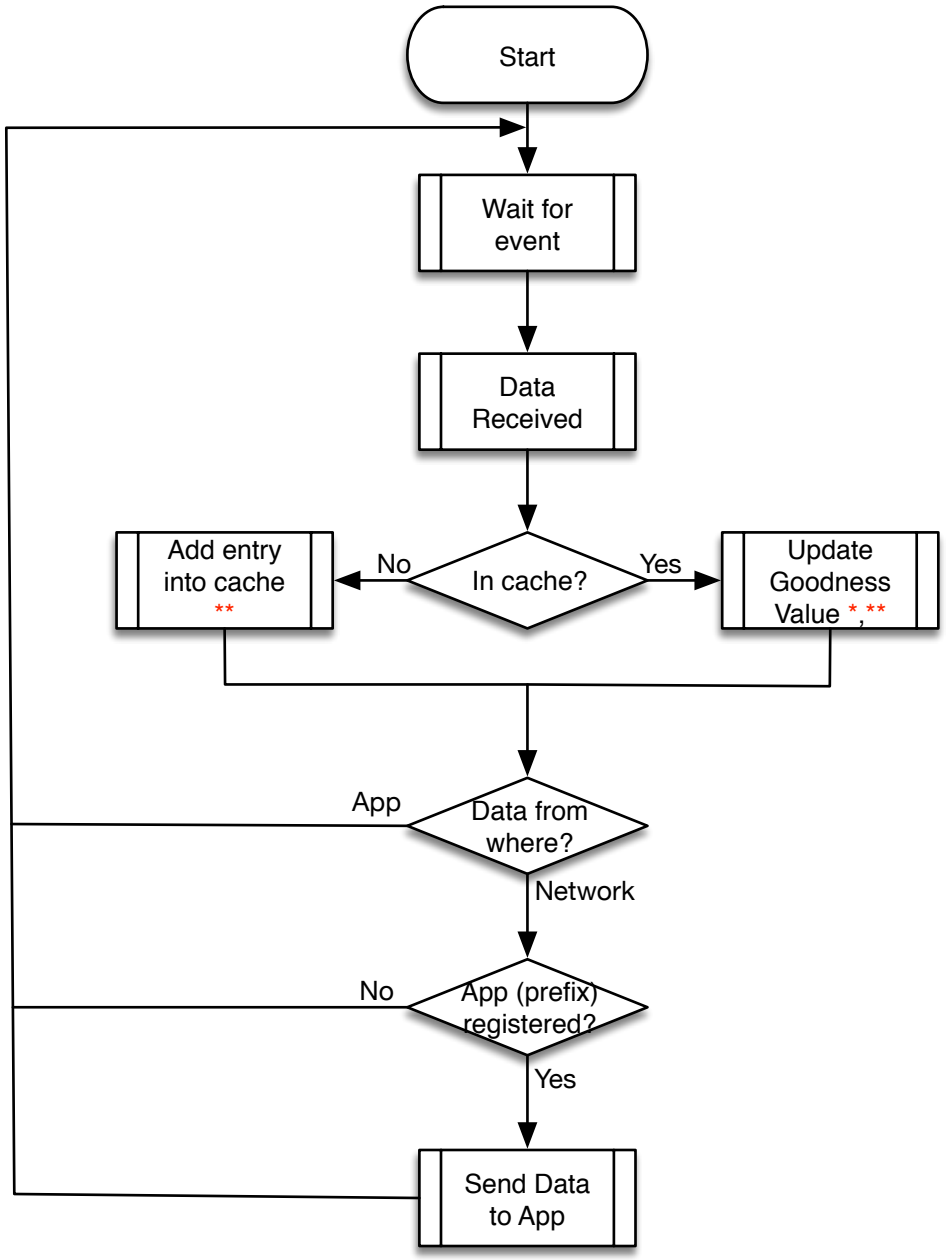

Keetchi
processDataMsg()
processFeedbackMsg()
processNewNeighbourList()
ageData()
CommMgr
determineChangeSignificance()
updateNeighbours()
DataMgr
updateCacheEntry()
recomputeGoodnessValue()
getCacheEntriesToSend()
ageCacheEntries()
ResourceMgr
determineResourceLimitationStatus()

Data Receipt

processDataMsg()



* Recompute and update Goodness value as follows

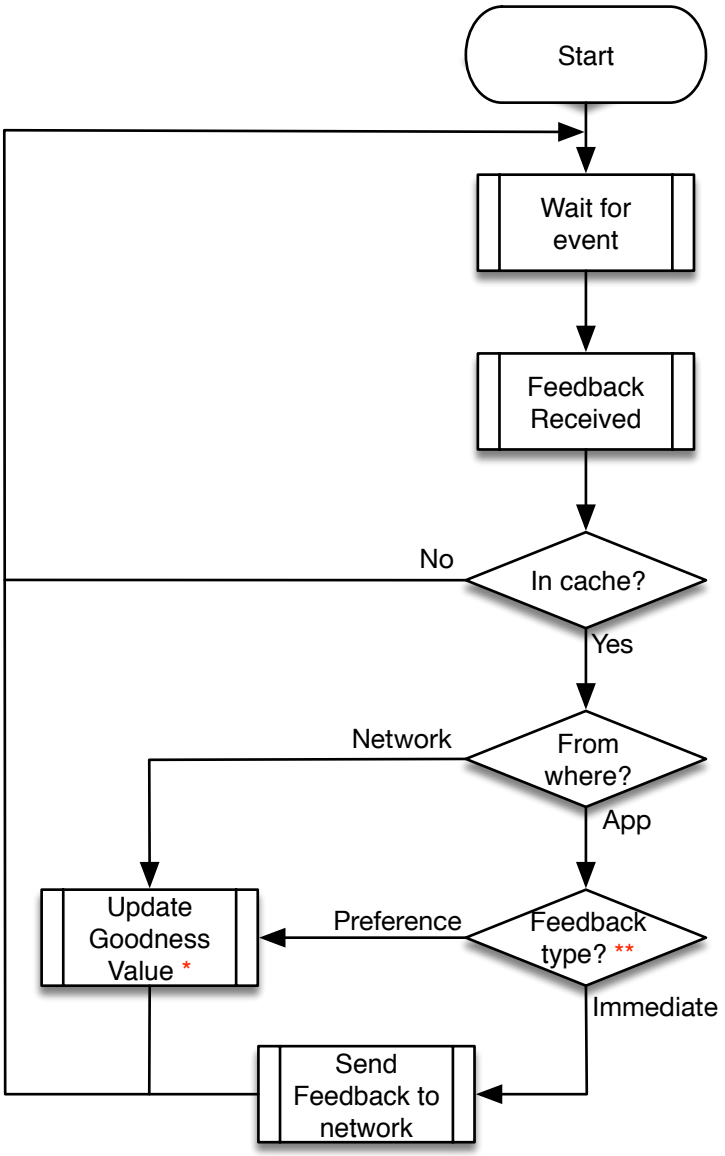
$$GV_{new} = GV_{old} \cdot \gamma + GV_{rcvd} \cdot (1 - \gamma)$$

GV = Goodness Value
GV_{old} = GV of current cache entry
GV_{rcvd} = GV of the received Data
GV_{new} = GV newly computed
γ = learning constant (0.0 - 1.0)

** Order entries in an ascending order of Goodness value. Ordering is required when selecting what Data to send (significant or non-significant change).

Feedback Receipt

processFeedbackMsg()



* Recompute and update Goodness value as follows

$$GV_{new} = GV_{old} \cdot \gamma + GV_{rcvd} \cdot (1 - \gamma)$$

GV = Goodness Value
GV_{old} = GV of current cache entry
GV_{rcvd} = GV of the received Data
GV_{new} = GV newly computed
γ = learning constant (0.0 - 1.0)

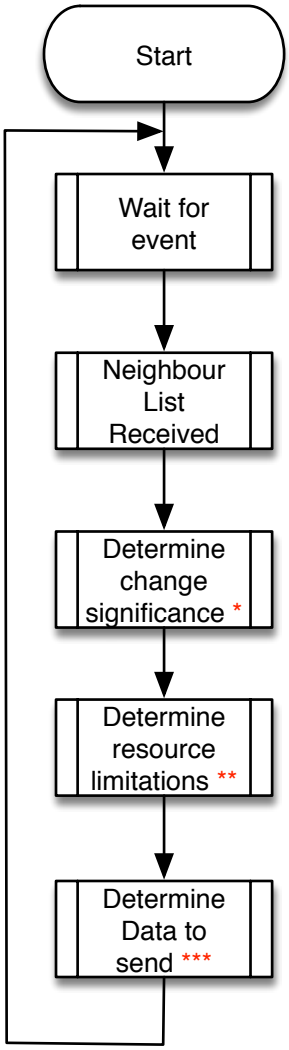
** Feedback types:

Immediate - A Feedback generated immediately by an App as a result of the receipt of a Data. The GV is the same as before, so no update done.

Preference - A Feedback generated due to user of an App indicating a like, dislike, etc. of the Data. GV is a value decided by the user, so has to be updated.

Neighbour List Receipt

processNewNeighbourList()

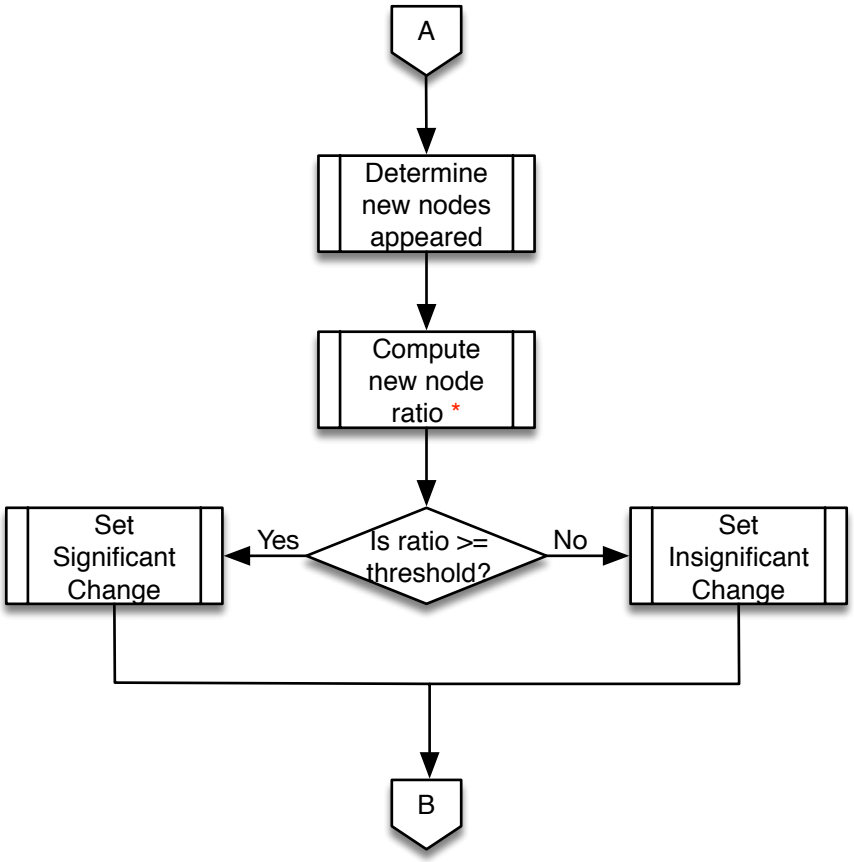


* Elaborated in a following page

** Currently consider as having unlimited resources

*** Elaborated in a following page

Determine Change Significance



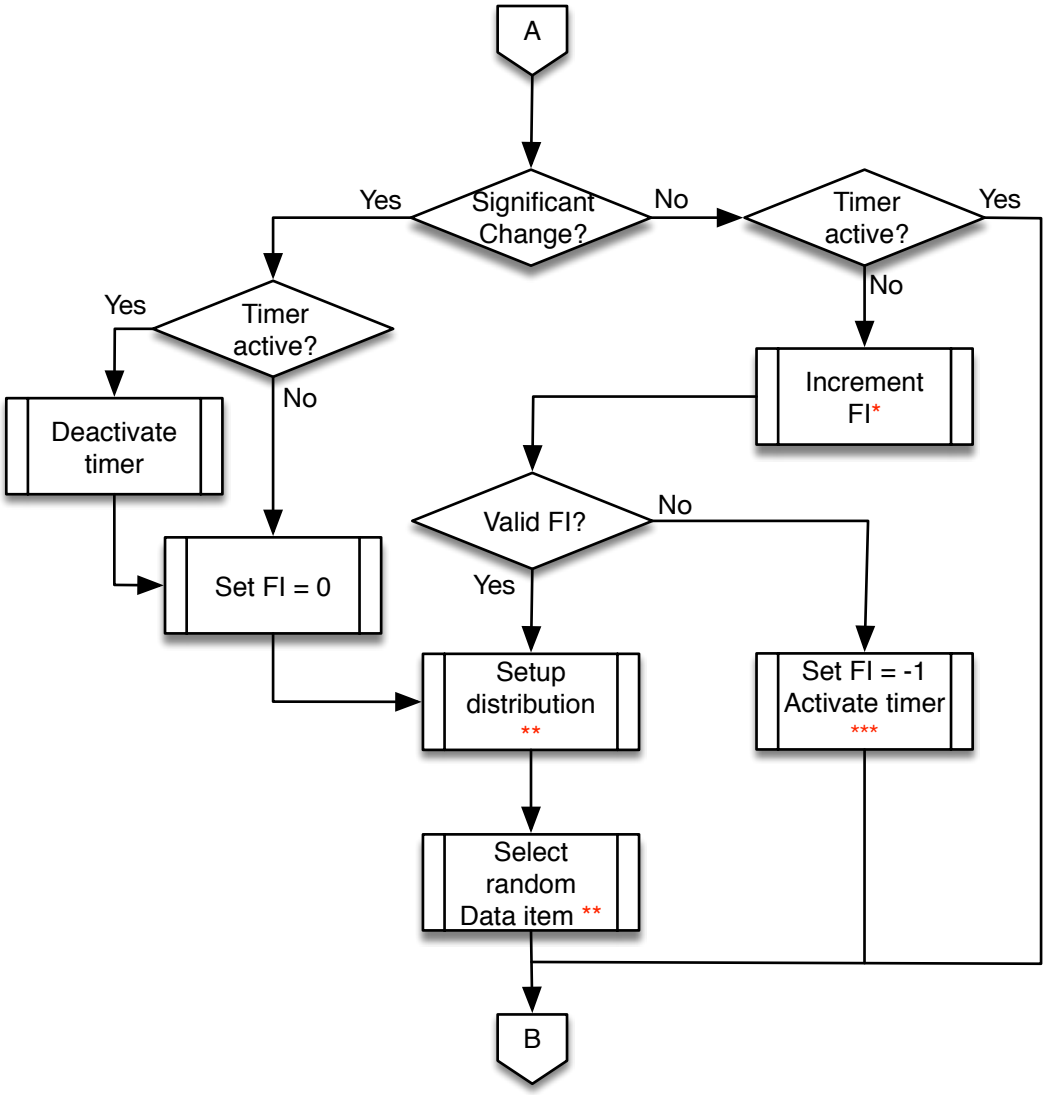
* ratio = D_a / D_n

D_a = new node arrivals

D_n = new neighbour list

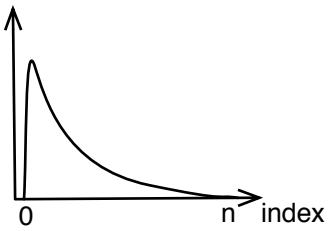
D = density, i.e., node count

Determine Data Item To Send

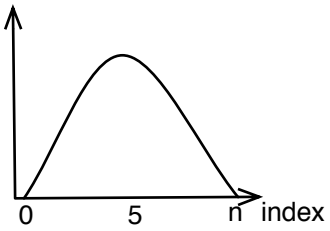


* FI = Focus Index
Index = cached data item index

** Probability distributions used to select random Data item



When FI = 0

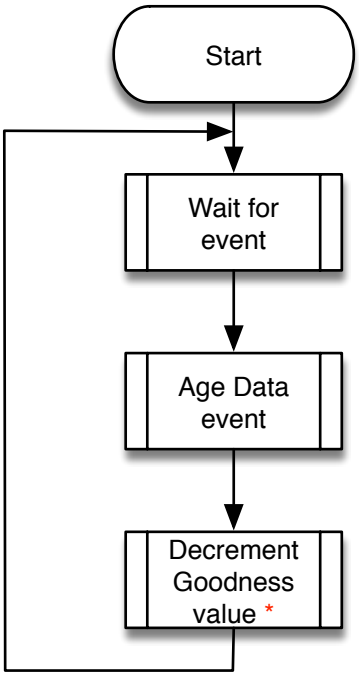


When FI = 5

*** When continuous **Insignificant Change** is received and the FI reaches end of cache size, no more Data are sent until a **Significant Change** is received or a **timer** is expired

Age Data

ageData()



* For every Data item in cache decrement Goodness Value by one