1. Bartendr Dynamic Programming:

   The energies required to transfer a frame at each of the 6 slots are the following: slot 1=1000 mJ, slot 2=1500 mJ, slot 3=700mJ slot 4=500mJ, slot 5=2000mJ and slot 6=110mJ. The tail time (time during which Radio-card is ON after sending a frame) is half of the slot time and Tail Energy is 600 mJ.

   $E_{3,6}$ = Optimal Energy to schedule 3 frames in 6 slots

Initialization
for $t = 1$ to M **do**
  $E_{0,t} = 0$
end for

Computing optimal schedules
for $k = 1$ to N **do**
  for $t = k$ to M **do**
    $E_{k,t} = \min_{\ell=k-1}^{t-1}(E_{k-1,\ell} + ESlot_{\ell+1})$
    $Last_{k,t} = \ell$ value for which the previous quantity was minimized
  end for
end for

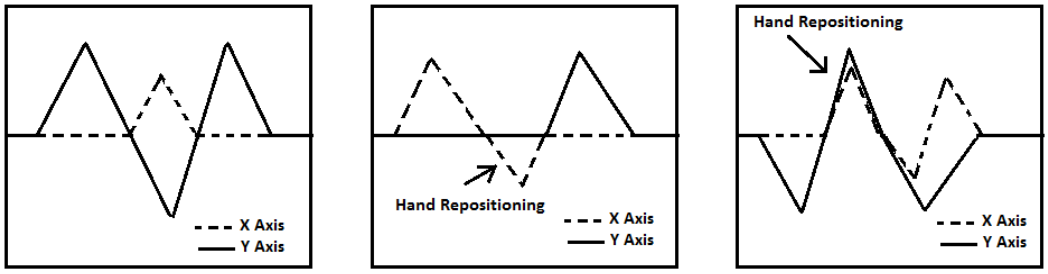
E	t=1	t=2	t=3	t=4	t=5	t=6
k=0	0	0	0	0	0	0
k=1	$E_{1,1}$=1600 $Last_{1,1}$=0	$E_{1,2}$=1600 $Last_{1,2}$=0	$E_{1,3}$=1300 $Last_{1,3}$=2	$E_{1,4}$=1100 $Last_{1,4}$=3	$E_{1,5}$=1100 $Last_{1,5}$=3	$E_{1,6}$=770 $Last_{1,6}$=5
k=2		$E_{2,2}$=3100 $Last_{2,2}$=1	$E_{2,3}$=2800 $Last_{2,3}$=2	$E_{2,4}$=1800 $Last_{2,4}$=3	$E_{2,5}$=1800 $Last_{2,5}$=3	$E_{2,6}$=1800 $Last_{2,6}$=3
k=3			$E_{3,3}$=3800 $Last_{3,3}$=2	$E_{3,4}$=3300 $Last_{3,4}$=3	$E_{3,5}$=3300 $Last_{3,5}$=3	$E_{3,6}$=2510 $Last_{3,6}$=5

Optimal Energy = 2510 mJ Optimal Slots 3,4,6

2. i)

We have not mentioned anything about hand repositioning in the question. So, students may or may not draw the axis plots for hand repositioning.

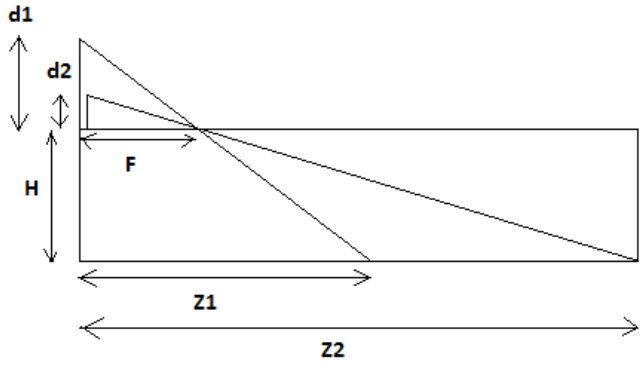
I have taken care of hand repositioning within charater. Somebody may take care of Inter-Character hand repositioning also.



ii) Edit distances b/w “ITILL” and “STILL”=1 & “ITILL” and “HILL”=2

But the right choice should be HILL because “H” (written in air) can be wrongly detected as “IT” by Phone point pen.

3. i)



$$Z_2 - Z_1 = (\frac{1}{d_2} - \frac{1}{d_1}) \times f \times H$$

Z2-2500= (1/10-1/25)*50*2000 = (15*50*2000)/250=6000

Z2=8500 units

ii) Any 3 of the following 4 points

a) The first source of control-flow graph ambiguity involves the class hierarchy.
b) Second, the possible values that each term in a comparison can take must be known to determine which paths through a program are feasible.

In addition, there are two last potential sources of ambiguity:

c) native code and
d) the reflection language feature.

iii) Typically, the bytecode that runs in a Dalvik virtual machine is from the file embedded in the app or the framework itself. For extensibility, Android provides a mechanism that can be

used to load and execute bytecode from an arbitrary source at run-time. Specifically, an app can leverage the DexClassLoader feature to load classes from embedded .jar and .apk files.

The dynamic loading of new class files could potentially change the code to run and thus reduce the effectiveness of static-analysis efforts. On the other hand, we cannot consider all apps with dynamic loading behavior malicious, because this behavior can be used legitimately. For example, apps can use this mechanism to update their functionality without reinstalling the app itself.

iv)

$$E_t = \begin{cases} P_i * (\frac{N}{f} - \frac{N*S}{B}) \\ \quad +P_a * \frac{N*S}{B} + E_{switch} & \text{if} \frac{N}{f} - \frac{N*S}{B} > Th_{idle} \\ P_a * \frac{N}{f} & \text{otherwise} \end{cases}$$

No. of Samples = N= (10*1024) / 1 =10240
f= 100 samples/second
S=1 KB
B=100KBPS
Th_{idle} = 100 ms

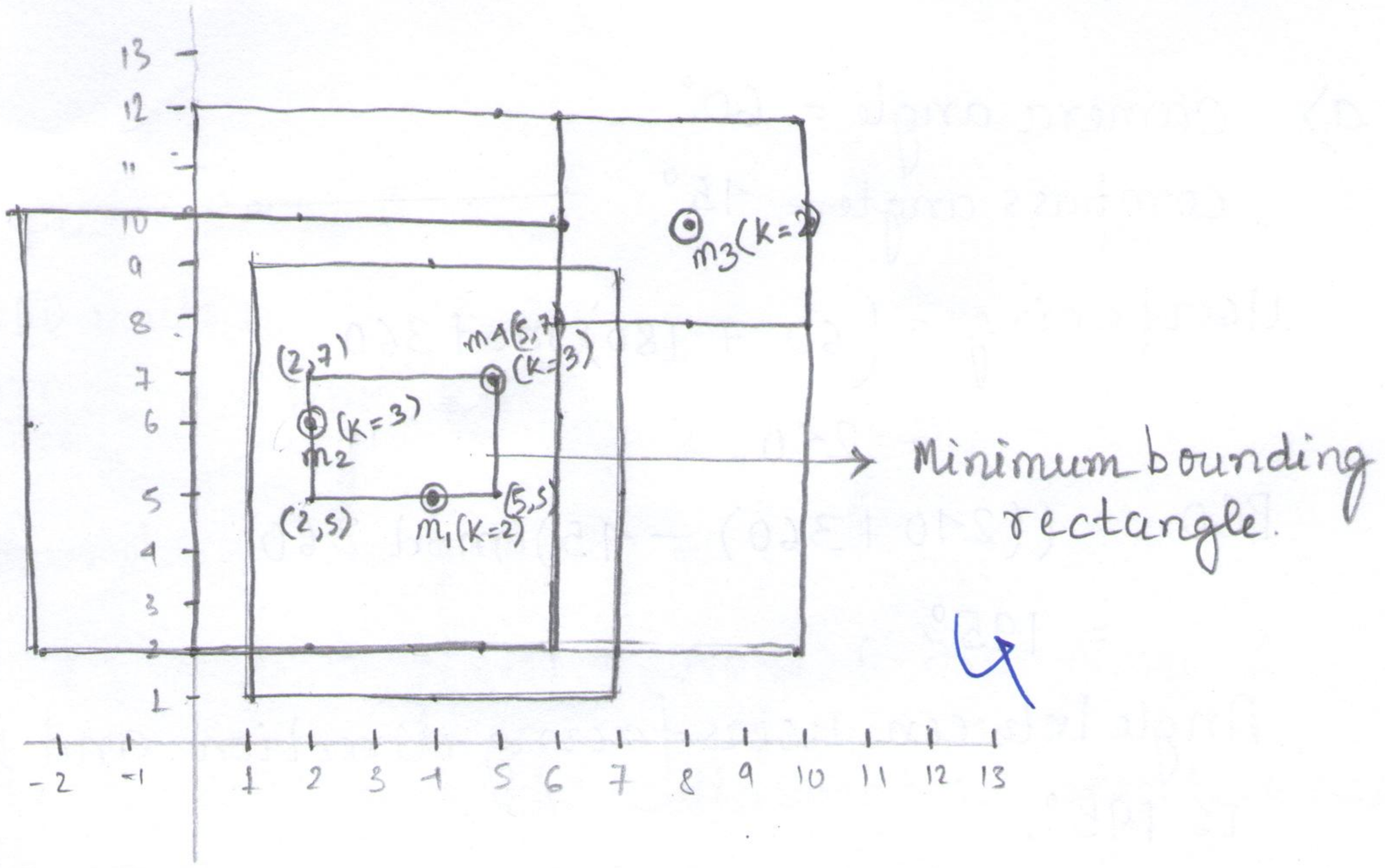
N/f =102.4 sec
(N*S)/B=10240/100 =102.4 sec

N/f-(N*S)/B =0 < Th_{idle}
So, E_t = 950 * 10^{-3} W* 102.4 sec = 97.28 Joule

v) The advantage of using a virtual machine is twofold – firstly, the app code is isolated from the core operating system, ensuring that should something go wrong, it's contained in an isolated environment and does not effect the primary OS.

And secondly, it allows for cross-compatibility, meaning even if an app is compiled on another platform (such as a PC, as is usually the case with developing mobile apps) , they can still be executed on the mobile platform using the virtual machine.

4 a)



m_1, m_3, m_4

b) m_1, m_2 and m_4 can be anonymized together as each message is present within each other's constraint box and forms a L-clique (L=3) and $m_1.k(2) \geq L$, $m_2.k(3) \geq L$ and $m_4.k(3) \geq L$

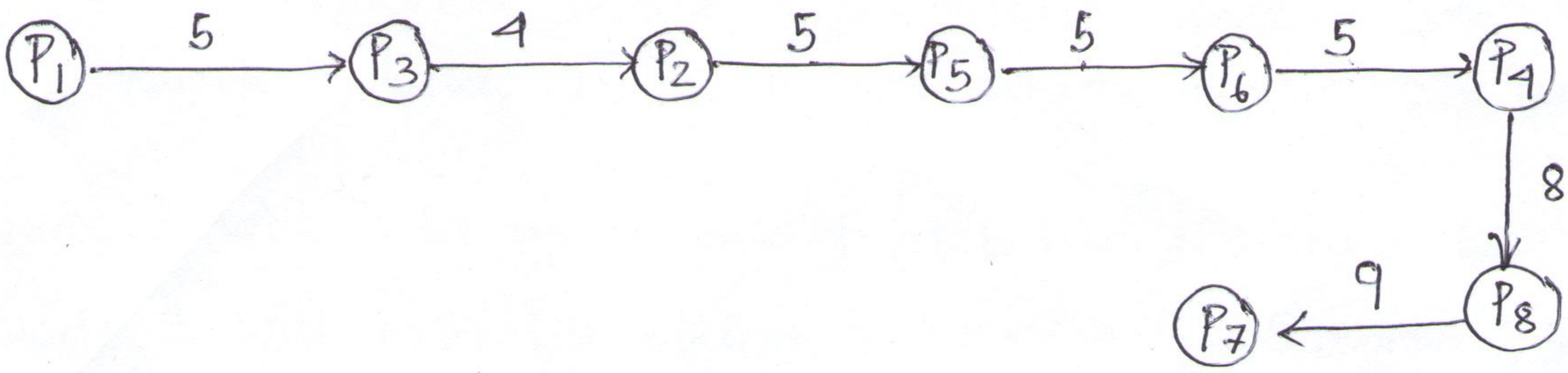
c) Dimension of the spatial cloaking box -

$x_s = 2$, $x_e = 5$

$y_s = 5$, $y_e = 7$

✓

5) Since the value of a is infinite, hence the walker will always choose the nearest waypoint. So the route will be -



The total cost is 41

6

6) a) camera angle = 60°

compass angle = 45°

Userfacing = (60 + 180) mod 360

= 240

PCO = ((240 + 360) − 45) mod 360

= 195°

Angle between users facing direction and her camera is 195°.

b) precision = $\frac{5}{8}$ = 0.625 3

recall = $\frac{5}{7}$ = 0.714 1,1

{People Inside ∩ Tagged by Tagsense} = $\{x_1, x_2, x_3, y_3, z_1\}$

7) a) Download size = 2000 KB

Cost = 1000 units.

TC = $\frac{1000}{2000}$ = 0.5 ~~k~~unit/KB

So the collaborators who qualify –

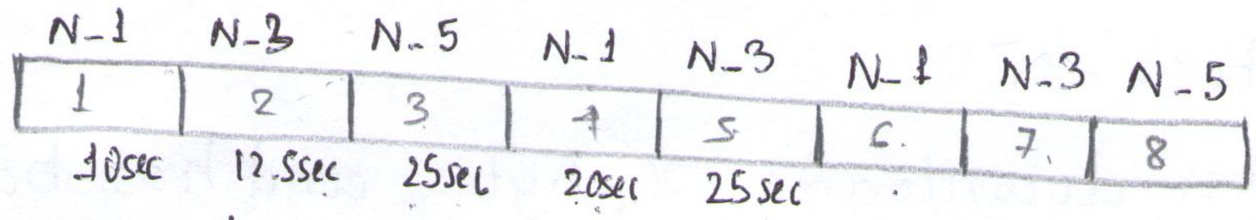
Node_1, Node_3, Node_4, Node_5 and Node_6

Based on the WNAN speed the collaborators selected are

Node_1, Node_3 and Node_5 (3) ??

b) The total file is divided into 8 chunks of equal size.

Each chunk is of size ~~125 kB.~~ 250 kB.



~~Bas~~

1st chunk is taken up by node_1

2nd chunk is taken up by node_3

3rd chunk is taken up by node_5 (5)

Chunk 1 takes 10 sec to download . cost = ~~50 units~~ 100 unit

Chunk 2 takes 12.5 sec to download cost = ~~37.5 units~~ 75 unit

Chunk 3 takes 25 sec to download. cost = ~~25 units~~ . 50 unit

Node_1 will take up chunk 4 as it has finished earliest cost = ~~100~~ 50

Node_3 will take up chunk 5 . cost = ~~37.5 units~~ 75 units

Node_1 will take up chunk 6. cost = ~~100~~ 50 units.

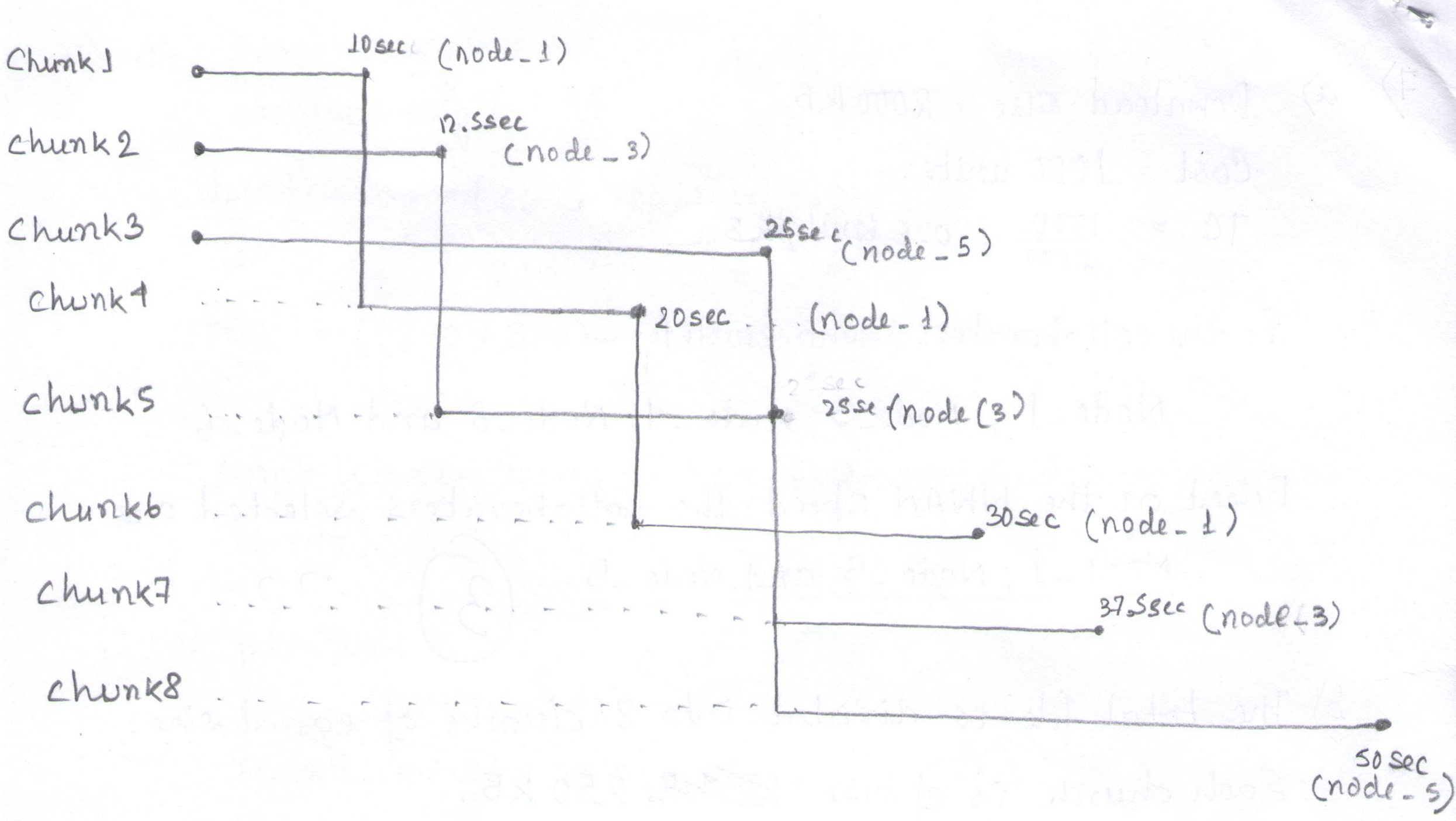
Node_3 will take up chunk 7. cost = ~~37.5 units~~ 75 units

Node_5 will take up chunk 8. cost = ~~25 units~~ . 50 units

Total cost = (~~100~~ 50 × 3) + (~~37.5~~ 75 × 3) + (50 × 2)

= ~~150 + 112.5 + 50~~ 300 + 225 + 100

= ~~312.5 units~~ . 625 units.



b) No of collaborators - N

each collaborator downloads x_i bytes and has bandwidth B_i

Total time taken to download the file:

$$\max \{x_i / B_i\} \quad \forall i$$

We minimize the total time subject to the constraints

$$\sum_{i=1}^{N} T C_i x_i \leq C$$

$$\sum_{i=1}^{N} x_i = F$$

$$x_i \geq 0, \quad i = 1, \ldots N$$

4

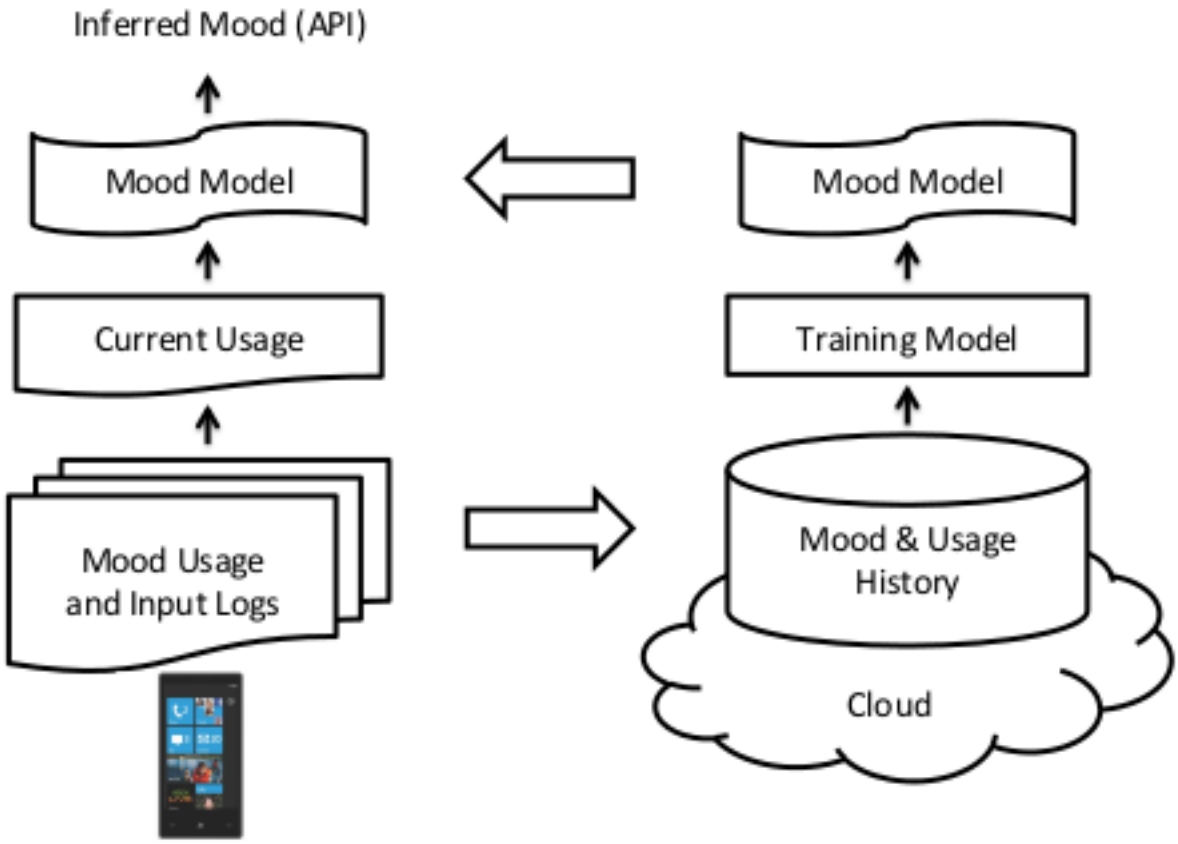


Figure 13: The Mood Inference Engine

8(a)

The mood inference engine consists of two software components one residing in the phone while other in the cloud. The phone side software collects smartphone usage logs and user mood labels, on behalf of the cloud. The cloud is responsible for training a predictive mood model using these data, which is provided back to the smartphone. By applying this model the phone is able to locally infer user mood without the aid of the cloud.

(b) To reduce the size of the feature space the system uses sequential forward selection algorithm.
In SFS, the system attempts to pick a subset Y of the feature table that will give the best regression. In the SFS algorithm, Y starts out as an empty set. SFS then iterates, finding the feature x which is not already in Y that provides the best fit to the data, minimizing the mean error of the fit. It then adds x to Y and continues to iterate. SFS will stop running when it reaches a local minimum; at this point, adding any of the remaining features will increase the error. Through this process, SFS appropriately selects the data's

most representative features, reducing the dimensionality.
8.c)
Limitation:
While personalized models report high accuracy, they require individual training over a long period of time.

Hybrid Mood Model

An ideal mood model would blend together the respective strengths of the personalized (high accuracy) and all user (no user training) modeling approaches. We investigate hybrid mood model design choices that combine a small amount of user specific training data with larger amounts of training data collected from the general user population. To test a hybrid approach, we fit a multi-linear regression model with a modified objective function that prioritizes reducing residual errors related to personalized training data above errors related to training data sourced from the rest of the population. As a result, the hybrid model is able to incorporate unique characteristics in how an individual user's smartphone data relate to their mood while also capturing the coarse-grain patterns that are common across all people. Through our experiments we find the accuracy of the hybrid model is naturally sensitive to the weighting, i.e., prioritization, placed on personalized data residual errors relative to data from the general population; we determine this weighting term empirically by performing a conventional grid parameter search that minimizes MSE error.

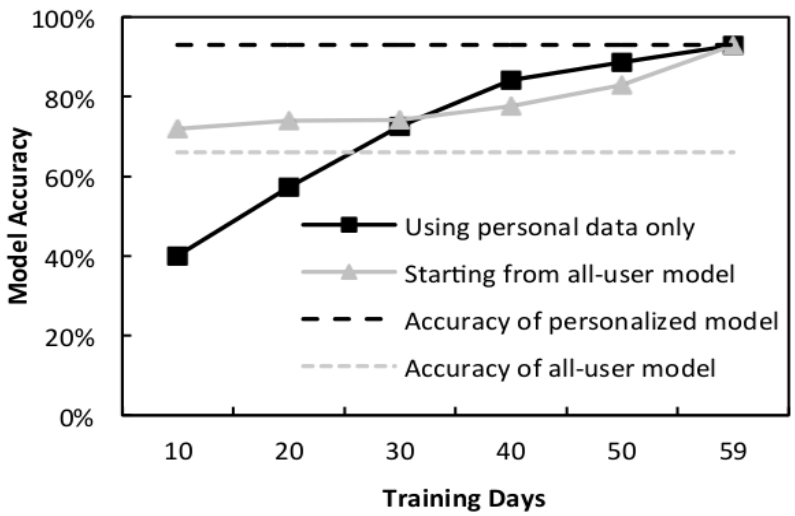


Figure 1: Pleasure training accuracy vs. training data size

Figure 1 examines the MSE accuracy of our hybrid model when incrementally trained with increasing amounts of user-specific training data. In this figure, we compare the hybrid approach with user training data to an incrementally trained personalized model

with no prior training data. We also represent the accuracy of the all-user model (66%) and personalized model (93%) as reference points. We find that even with only 10 days of personalized training data, the hybrid model has higher MSE accuracy than the incremental personalized by 31% and the all user model by 6%. After 30 days, the personalized model outperforms the hybrid model and can be used with greater than 75% accuracy.