Two problems regarding Hyperloglog on vehicle counting.

Table of notations:

Notation for input:	
S	Input stream of data elements
n	Number of distinct data elements
Notation for Hyperloglog:	
m	Number of substreams in S
p	Precision argument, p = $\log_2(m)$
L	Length of hash value for each data element, typically value is 32
Notation for Linear counting:	
V_0	Fraction of empty bits of bitmap
t	Load factor, $E(V_0) = e^{-t}$.

Memory comparison with same level of accuracy

1. For Hyperloglog, to reduce the large variability of using a single measurement, stochastic averaging is used. To that end, the input stream of data elements S is divided into m substreams S_i of roughly equal size, using the first p bits of the hash values, where $m = 2^p$.

   The error rate for Hyperloglog is

   $$\frac{1.04}{\sqrt{m}}$$

   As this require the use of m registers, the memory consumption is

   $$log_2(L) * m$$

   where L is the length of hash value each data element generates.

2. For linear counting, the error rate is

   $$\frac{e^t - t - 1}{2n}$$

   where t is the load factor, $E(V_0) = e^{-t}$.

   The memory consumption can be estimated as

   $$m = \frac{n}{t}$$

3. With equations in 1 & 2, let's do some rough comparison.
   I found several number on Internet:
   The number of vehicles in US is around 300M.
   The number of vehicles in NY city is around 3M.

Suppose n = 100k.

a. For Hyperloglog
    i. To achieve an **error rate of 0.01**, with a typical 32bit hash, we need approximately **6.7KB memory**.
    ii. Note that 32bit hash does not support an error rate of 0.001, as the range of p can only be [4, 16]. The p value for an error rate of 0.001 is 21. So, we use 64bit hash instead.
    So, To achieve an **error rate of 0.001** with a 64bit hash, we need approximately **800KB memory**.
b. For Linear counting
    i. To achieve an **error rate of 0.01**, we have t approximately at 7.6. We need approximately **1.6KB memory**.
    ii. To achieve an **error rate of 0.001**, we have t approximately at 5.3. We need approximately **2.4KB memory**.

Hyperloglog can use very limited memory to estimate **very large datasets (above billions)** as the size of memory needed is mainly determined by the required accuracy. Unfortunately, the case of vehicle counting does not fall into that category. **In practical variant of Hyperloglog algorithm, for small cardinalities, i.e., n < 2.5m, Linear counting is used [1]**.

Hyperloglog in persistent traffic counting

Given a certain number of measurement periods, the persistent traffic is defined as the set of

vehicles that pass the location in all those periods.

Suppose we have 2 measurement periods, A and B. A has 10K vehicles, B has 15K vehicles, A∩B has 7K vehicles. In this case, the persistent traffic is 7K.

However, with Hyperloglog, the best we can get by using the “min length of the leading 0s” is the 10K, which is much larger than 7K. What I mean in last meeting is that there seems no way for Hyperloglog to estimate the 7K persistent traffic accurately in such a case.

[1] Heule, Stefan, Marc Nunkesser, and Alexander Hall. "HyperLogLog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm." *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, 2013.