Traffic measurements are a critical component for the control and engineering of a network. Being able to provide a good service quality and planning over fail over strategies requires understanding the spatial flow of traffic through the domain. It's specially important to know:

1. Which are the ingress points
2. Where is the traffic headed
3. What trajectory it takes

Two types of traffic monitoring methods are discussed on the paper using indirect and direct traffic measurements. In indirect methods the traffic is observed at a specific point, for example in the ingress device, and in order to infer where the traffic is heading statistics are calculated based on source destination pairs and the state of the routing protocol. Of course those methods might suffer from accuracy issues and mis-configurations are easy to happen due to how complicated the routing protocols are. That's why direct measurements methods come to the rescue which don't rely on the network model but instead directly take traffic measurements at multiple points in the network.

The paper proposes trajectory sampling, a direct method being able to monitor the way the packets are in the network. Overall 2 hash functions are used and a sampling tactic applied on all the packets entering the network. The main idea is to use a hash function in the ingress router in order to decide whether to sample the packet or not. The decision can be based on anything that fits the application needs. The same function is applied on every routing node, so a packet sampled on the first router is sure to be sampled on all the way until the egress router. In order to distinguish between the sampled packets stored another hash function is used to choose just the minimum amounts of bits which give us a unique with high probability representation of the original packet.

The second function picks up the packet headers and stores the bits providing the higher entropy. The writers provide as with a figure indicating the amount of entropy each packet header field provides (figure 1). Furthermore the bits extracted that way provide an effective way of compressing the packet without having to store information about the frequency of the repeated packets and symbols. It's useful here to note that additional info about the packet might be useful to be stored. That's the job of the ingress router while others only need to store info about the labels (compressed packet)
The proposed method has various advantages since there is no need to store the routing state or the network topology or provide special treatment to multicast packets. Although someone would thought that its sufficient, it suffers for identical packets. The paper indicates a way to distinguish unambiguous trajectories caused by these identical packets. As they say the subgraph of a trajectory is unambiguous if each connected component of the subgraph is either a source tree or a sink tree having as many inbound links as the outbound links. In figure 2 there are still 3 graphs F,G,H which remain unambiguous.
Exactly identical packets remain an open problem in this paper however for those not exactly the same a stricter hash function can reduce and make collisions a rare phenomenon. From figure 3 it can be observed that the high entropy bits of the header might not be enough however, taking some bytes from the packet payload can reduce the collision probabilities to arbitrary small amounts. Increasing the amounts of byte above 40 didn't offer any improvement cause the packets where indeed identical.

After justifying and validating the results the paper focuses on the implementation issues. The writers argue that the implementation cost is acceptable even for the highest interface speeds. Each device must be capable of computing the sampling and the identification hash however those two computations can be made in parallel as sown in the figure below.
To sum up, trajectory sampling offers a method of direct observation of the traffic without requiring additional memory accesses and knowledge of the network and the routing protocols. It may fail to distinguish the route of identical packets measured in the same time interval however it provides a great way of shielding the system against DDoS attacks since the trajectory the packets take can be observed. Furthermore, one may add control packets o the network while appending specific header values so that the packets are always sampled. That way one may verify or debug specific routes. In the case of tunneling the hash identification function needs to be smart enough to distinguished between the real packet header and the tunneled ones.