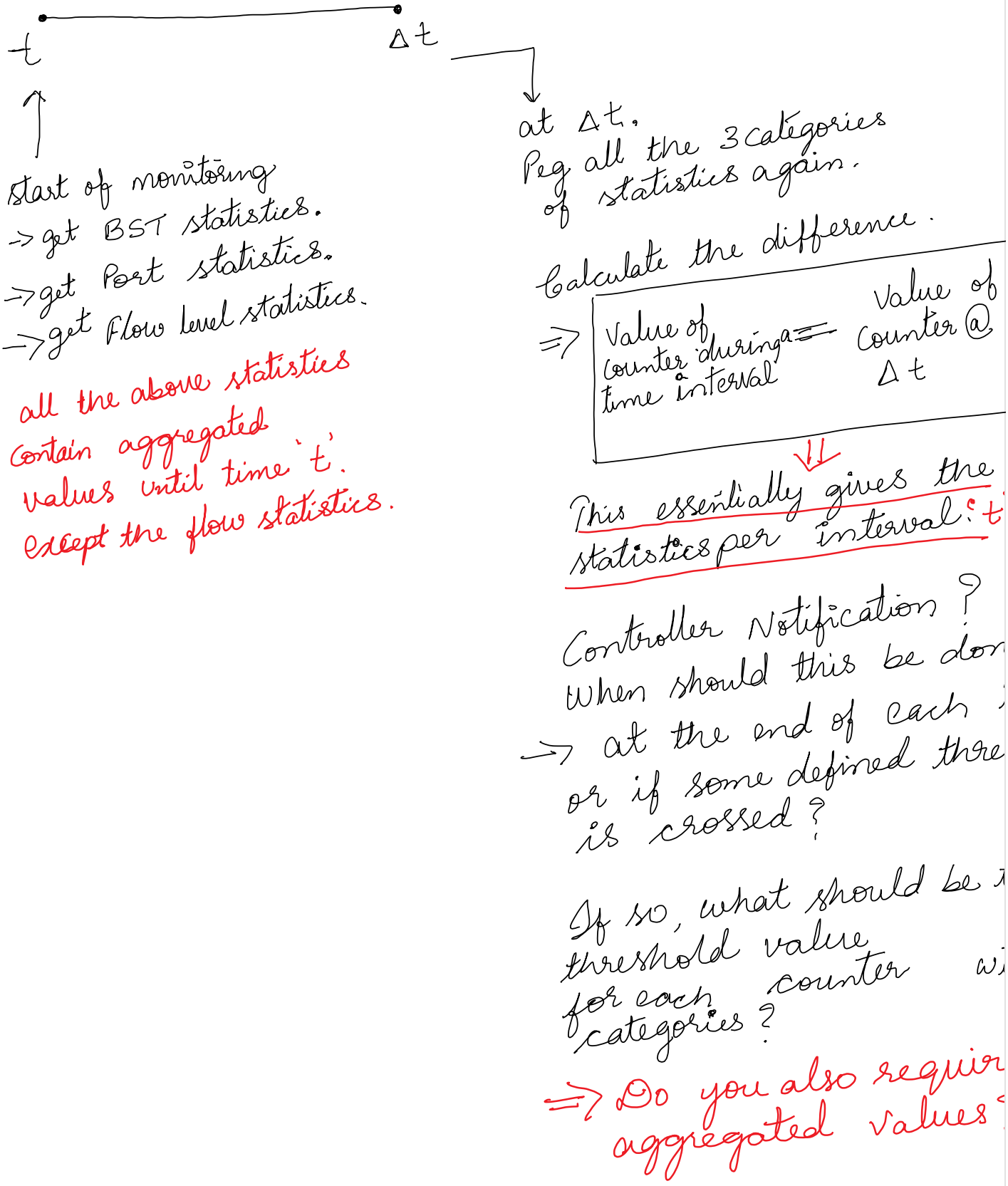
Counters collected and State Information to be reported.

Wednesday, January 18, 2017 3:50 PM



Following port level statistics and BID statistics are considered at the moment.

```
opennsl_spl_snmpIfInUcastPkts, /* 0 */              /* rx_packets */
opennsl_spl_snmpIfInNUcastPkts, /* 1 */
opennsl_spl_snmpIfOutUcastPkts, /* 2 */             /* tx_packets */
opennsl_spl_snmpIfOutNUcastPkts, /* 3 */
```

```
  opennsl_spl_snmpIfInOctets, /* 4 */                /* rx_bytes */
  opennsl_spl_snmpIfOutOctets, /* 5 */               /* tx_bytes */
  opennsl_spl_snmpIfInErrors, /* 6 */                /* rx_errors */
  opennsl_spl_snmpIfOutErrors, /* 7 */               /* tx_errors */
  opennsl_spl_snmpIfInDiscards, /* 8 */              /* rx_dropped */
  opennsl_spl_snmpIfOutDiscards, /* 9 */             /* tx_dropped */
  opennsl_spl_snmpEtherStatsMulticastPkts, /* 10 */  /* Multicast */
  opennsl_spl_snmpEtherStatsCollisions, /* 11 */     /* collisions */
  opennsl_spl_snmpEtherStatsCRCAlignErrors, /* 12 */ /* rx_crc_errors */
  opennsl_spl_snmpIfInMulticastPkts, /* 13 */        /* rx multicast */
  opennsl_spl_snmpIfInBroadcastPkts, /* 14 */        /* rx broadcast */
  opennsl_spl_snmpIfInUnknownProtos, /* 15 */        /* rx unknown protos */
  opennsl_spl_snmpIfOutMulticastPkts, /* 16 */       /* tx multicast */
  opennsl_spl_snmpIfOutBroadcastPkts /* 17 */        /* tx broadcast */


t_counter_t id_list[MAX_COUNTERS] = {
  {opennslBstStatIdUcast,           "opennslBstStatIdUcast"},
  {opennslBstStatIdMcast,           "opennslBstStatIdMcast"},
  {opennslBstStatIdPriGroupShared,  "opennslBstStatIdPriGroupShared"},
  {opennslBstStatIdPriGroupHeadroom, "opennslBstStatIdPriGroupHeadroom"}/*,
```

State information maintained is as below:

```
struct monitoring_state_t
{
    struct fp_flow_stats_t flow_stats_val;      //flow level statistics populated by the monitoring process at the beginning
    struct fp_flow_stats_t flow_stats_val_delta;//Populated by the monitoring process at the end of each interval.
    struct fp_flow_stats_t flow_stats_val_interval; //flow level statistics populated for an interval by the monitoring proce
    int port_stats_val[MAX_STAT_COUNTERS];      //port level statistics populated by the monitoring process at the beginning
    int port_stats_val_delta[MAX_STAT_COUNTERS];//port level statistics populated by the monitoring process at the end of eac
    int port_stats_val_interval[MAX_STAT_COUNTERS];//port level statistics for an interval populated by the monitoring proces
    struct bst_val_counters bst_stats;          //BST statistics populated by the monitoring process at the beginning of each
    struct bst_val_counters bst_stats_delta;    //BST statistics populated by the monitoring process at the end of each inter
    struct bst_val_counters bst_stats_interval; //BST statistics for an interval populated by the monitoring process at the e
};
```

Collected at time t.

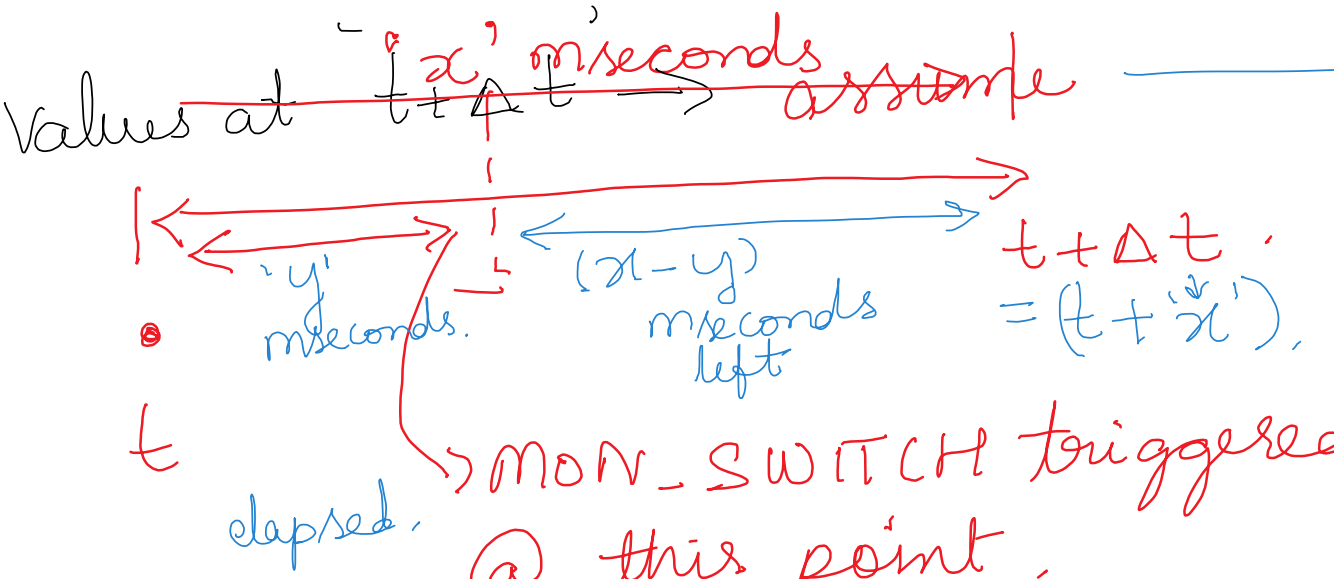
Collected at time t+Δt } Both are aggregated values.

Contains diff of (t) - (t+Δt)
⇒ values of counters during a particular interval.

So which of these information should be transferred as 'state' information when we do 'monitoring switch' operation.

In my opinion
Values at 't' → considered as

previously obtained values.



So when the Monitoring is to "Switch S2".

It shall have the following.

→ Values collected @ 't'

→ Values collected @ 't+'

When 'S2' receives these Va

Eg starts monitoring,
It would, for the first tim
peg the counter values aft

NOTE: 'counters' during the ti
are not accounted.
Also, Time it could requi
depending on the load on s

⇒ We need to consider Ti
counters at time (t+Δ
So, for the first time on
values at 't' is taken,
values at (t+Δt) is ©

Σ (countervalues at (t+y) in secon

obtained from 'S 1'

Few questions here:

→ should I actually calcu
per interval of time b/w
and if the calculated s
some threshold?

↳ what is the
the controlle
statistics?

→ what should be the poll
i.e. "Δt" interval?

if it is large → Monito
tran

if it is too small → to
is
tr
c

what happens if a flow
for a few milliseconds/
it takes a few millisecond
monitoring on a different
resources?

→ In this case, we

any state informati

In my opinion,

flow monitoring is requi

→ Billing purposes.

→ Bandwidth utilization cal

→ Buffer tuning.

↳ This is done on a particular swi based on buffer consumption by the per port on per