

在Payless代码中有计算Utilization的部分。

当floodlight Controller接收到STATS_REPLY消息时，会计算Utilization并在DEBUG中输出Utilization的数值。

```
case STATS_REPLY:
    OFStatisticsReply statReply = (OFStatisticsReply) msg;
    ArrayList<OFStatistics> statList = (ArrayList<OFStatistics>) statReply.getStatistics();
    logger.debug("Received Stat Reply " + statList.size());
    if (statList != null && statList.size() > 0)
    {
        OFFlowStatisticsReply fstatReply = (OFFlowStatisticsReply) statList.get(0);
        //duration = fstatReply.getDurationSeconds() + (fstatReply.getDurationNanoseconds() / 1e9);
        //duration = matchedFlow.getScheduleTimeout() / 1000.0;
        duration = (double) fstatReply.getDurationSeconds() + (double) fstatReply.getDurationNanoseconds() / 1e9
            - matchedFlow.getDuration();

        byteCount = fstatReply.getByteCount() - matchedFlow.getMatchedByteCount();
        logger.debug("Matched Flow Prev. Byte Count = " + matchedFlow.getMatchedByteCount());
        logger.debug("Stat reply, Del-byte = " + byteCount);
        if (byteCount < MIN_SCHEDULE_BYTE_THRESHOLD)
        {
            int oldTimeout = matchedFlow.getScheduleTimeout();
            int newTimeout = Math.min(matchedFlow.getScheduleTimeout() * SCHEDULE_TIMEOUT_AMPLIFY_FACTOR,
                MAX_SCHEDULE_TIMEOUT);
            matchedFlow.setScheduleTimeout(newTimeout);
            schedule.updateTimeout(oldTimeout, newTimeout, matchedFlow);
            if (schedule.getAction(newTimeout) == null)
            {
                ScheduledExecutorService ses = threadPool.getScheduledExecutor();
                SingletonTask action = new SingletonTask(ses, new PollSwitchWorker(newTimeout, this));
                action.reschedule(newTimeout, TimeUnit.MILLISECONDS);
                schedule.addAction(newTimeout, action);
            }
        }
        else if (byteCount > MAX_SCHEDULE_BYTE_THRESHOLD)
        {
            int oldTimeout = matchedFlow.getScheduleTimeout();
            int newTimeout = Math.max(matchedFlow.getScheduleTimeout() / SCHEDULE_TIMEOUT_DAMPING_FACTOR,
                MIN_SCHEDULE_TIMEOUT);
            matchedFlow.setScheduleTimeout(newTimeout);
            schedule.updateTimeout(oldTimeout, newTimeout, matchedFlow);
            if (schedule.getAction(newTimeout) == null)
            {
                ScheduledExecutorService ses = threadPool.getScheduledExecutor();
                SingletonTask action = new SingletonTask(ses, new PollSwitchWorker(newTimeout, this));
                action.reschedule(newTimeout, TimeUnit.MILLISECONDS);
                schedule.addAction(newTimeout, action);
            }
        }
        matchedFlow.setMatchedByteCount(fstatReply.getByteCount());
        matchedFlow.setDuration((double) fstatReply.getDurationSeconds()
            + (double) fstatReply.getDurationNanoseconds() / 1e9);
    }
    break;
}
```

double utilization = (double) byteCount / duration;
logger.debug("Instant utilization = " + utilization);

但在Payless程序运行时，却没有Utilization的数值的输出：

```
mn命令:sudo mn --controller=remote,ip=127.0.0.1,port=6633
history:
[sudo] password for tangjixing:
*** Creating network
*** Adding controller
*** Adding hosts:
```

```

h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> exit
*** Stopping 1 controllers
c0
*** Stopping 2 terms
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 371.975 seconds

```

iperf命令:

```

h1 iperf -s
h2 iperf -c 10.0.0.1 -i -t 4s

```

floodlight DEBUG输出 :

```

20:18:24.927 [New I/O server worker #1-3] INFO n.f.forwarding.Forwarding - dl_vlan|dl_vlan_pcp|nw_tos|tp_dst|tp_src
20:18:24.927 [New I/O server worker #1-3] DEBUG n.f.routing.ForwardingBase - Route [id=RouteId [src=00:00:00:00:00:00:00:01 dst=00:00:00:00:00:00:00:01],
20:18:24.932 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Intercepted FlowMod Message:OFFlowMod [actionFactory=org.openflow.protoc
20:18:24.934 [New I/O server worker #1-3] INFO n.f.forwarding.Forwarding - dl_vlan|dl_vlan_pcp|nw_tos|tp_dst|tp_src
20:18:24.935 [New I/O server worker #1-3] DEBUG n.f.routing.ForwardingBase - Route [id=RouteId [src=00:00:00:00:00:00:00:01 dst=00:00:00:00:00:00:00:01],
20:18:24.935 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Intercepted FlowMod Message:OFFlowMod [actionFactory=org.openflow.protoc
20:18:24.936 [New I/O server worker #1-3] INFO n.f.forwarding.Forwarding - dl_vlan|dl_vlan_pcp|nw_tos|tp_dst|tp_src
20:18:24.936 [New I/O server worker #1-3] DEBUG n.f.routing.ForwardingBase - Route [id=RouteId [src=00:00:00:00:00:00:00:01 dst=00:00:00:00:00:00:00:01],
20:18:24.937 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Intercepted FlowMod Message:OFFlowMod [actionFactory=org.openflow.protoc
20:18:29.957 [New I/O server worker #1-3] INFO n.f.forwarding.Forwarding - dl_vlan|dl_vlan_pcp|nw_tos|tp_dst|tp_src
20:18:29.957 [New I/O server worker #1-3] DEBUG n.f.routing.ForwardingBase - Route [id=RouteId [src=00:00:00:00:00:00:00:01 dst=00:00:00:00:00:00:00:01],
20:18:29.958 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Intercepted FlowMod Message:OFFlowMod [actionFactory=org.openflow.protoc
20:18:29.962 [New I/O server worker #1-3] INFO n.f.forwarding.Forwarding - dl_vlan|dl_vlan_pcp|nw_tos|tp_dst|tp_src
20:18:29.962 [New I/O server worker #1-3] DEBUG n.f.routing.ForwardingBase - Route [id=RouteId [src=00:00:00:00:00:00:00:01 dst=00:00:00:00:00:00:00:01],
20:18:29.963 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Intercepted FlowMod Message:OFFlowMod [actionFactory=org.openflow.protoc
20:18:40.126 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - Flow Removed from sw = 1, byte counts = 25267624488 source = 10.0.0.2:0 des
20:18:40.127 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Removed flow: Switch ID = 1, In Port = 2, Source IP = 10.0.0.2, Destination IP =
20:18:40.127 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - [FLOW_MOD] Checkpoint = 1454156320127
20:18:40.127 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - nSwitches = 0
20:18:40.128 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - ****Overhead****
20:18:40.128 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - Flow Removed from sw = 1, byte counts = 47910774 source = 10.0.0.1:0 dest =
20:18:40.128 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - Removed flow: Switch ID = 1, In Port = 1, Source IP = 10.0.0.1, Destination IP =
20:18:40.128 [New I/O server worker #1-3] DEBUG n.f.netmonitor.NETMonitor - [FLOW_MOD] Checkpoint = 1454156320128
20:18:40.128 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - nSwitches = 0
20:18:40.128 [New I/O server worker #1-3] INFO n.f.netmonitor.NETMonitor - ****Overhead****

```