

TABLE OF CONTENTS

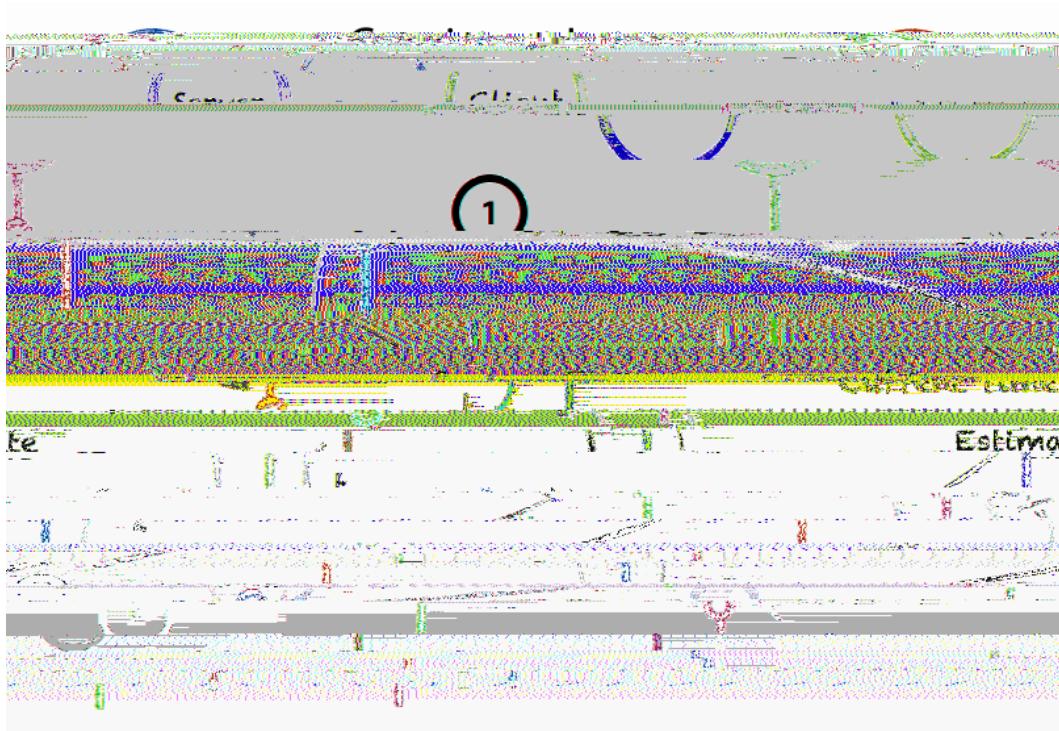
<i>Cristians Algorithm!</i>	6
<i>The Forder Algorithm!</i>	9
<i>Distributed Forder Algorithm!</i>	II
<i>Implementation!</i>	16
<i>Results</i>	

<i>E periment #1: Base Time Record Set!</i>	34
<i>E periment #2: Set up the Network!</i>	43
<i>E periment #3: Set up the Connect!</i>	50
<i>MRFo "o er Class!</i>	54
<i>MRTimeRecord Class!</i>	63
<i>MRTimeUpdateMessage Class!</i>	71
<i>MRNet ork Class!</i>	73
<i>MRCristian ork Class!</i>	76
<i>MRCristian ork Model Class!</i>	78
<i>MRNet orkNode Class!</i>	82
<i>MRNet orkDirector Class!</i>	84
<i>MRNet orkModelProtocol Class!</i>	87
<i>MRNormalDistr ork Model Class!</i>	88
<i>MRSimulation Class!</i>	90
<i>MRNet orkSimulation Class!</i>	94
<i>MRUni edAnomal Simulation Class!</i>	96
<i>Main!</i>	102
<i>Main!</i>	107
<i>MRLeanTimeRecord Class!</i>	113
<i>MRWiFi Class!</i>	115

LOGICAL TIME SYNCHRONIZATION IN
DISTRIBUTED NETWORKS WITH
VOLATILE LATENCY

Previous Work

CRISTIAN'S ALGORITHM



Cristian's Algorithm assumes latency is relatively consistent, so that dividing the total round-trip message time, t_r , in half will result in an accurate estimate of network latency.

It does not scale well:

Accuracy declines with increased latency volatility:

It is not naturally distributed:

It's a one-way protocol:

Very accurate in certain conditions:

Strong resistance to network volatility:

Adaptable to distributed networks:

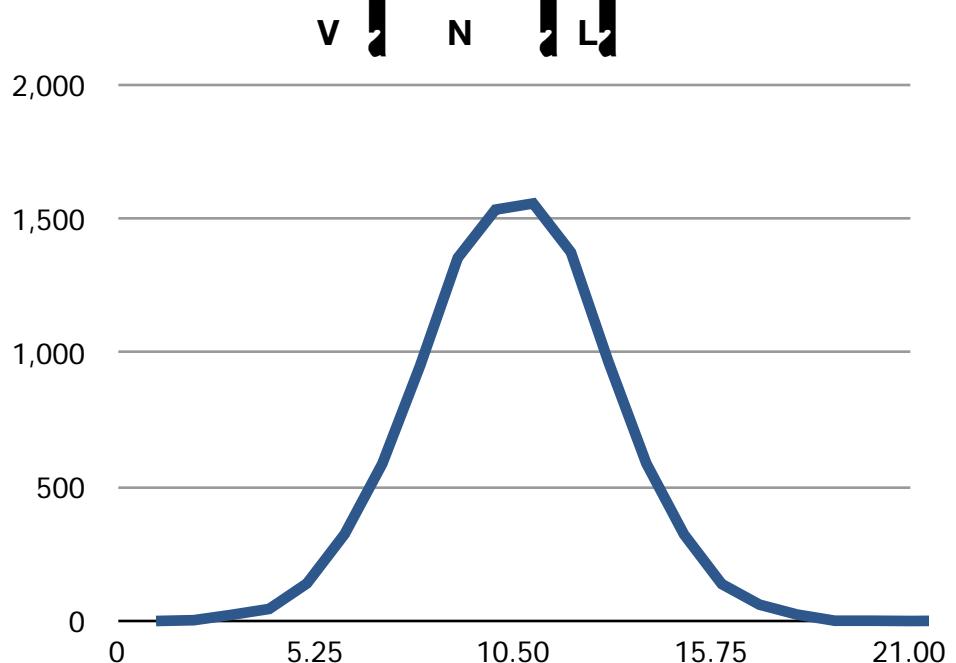
Adaptable to changes in overall network latency behavior:

Works well in both low and high latency volatility:

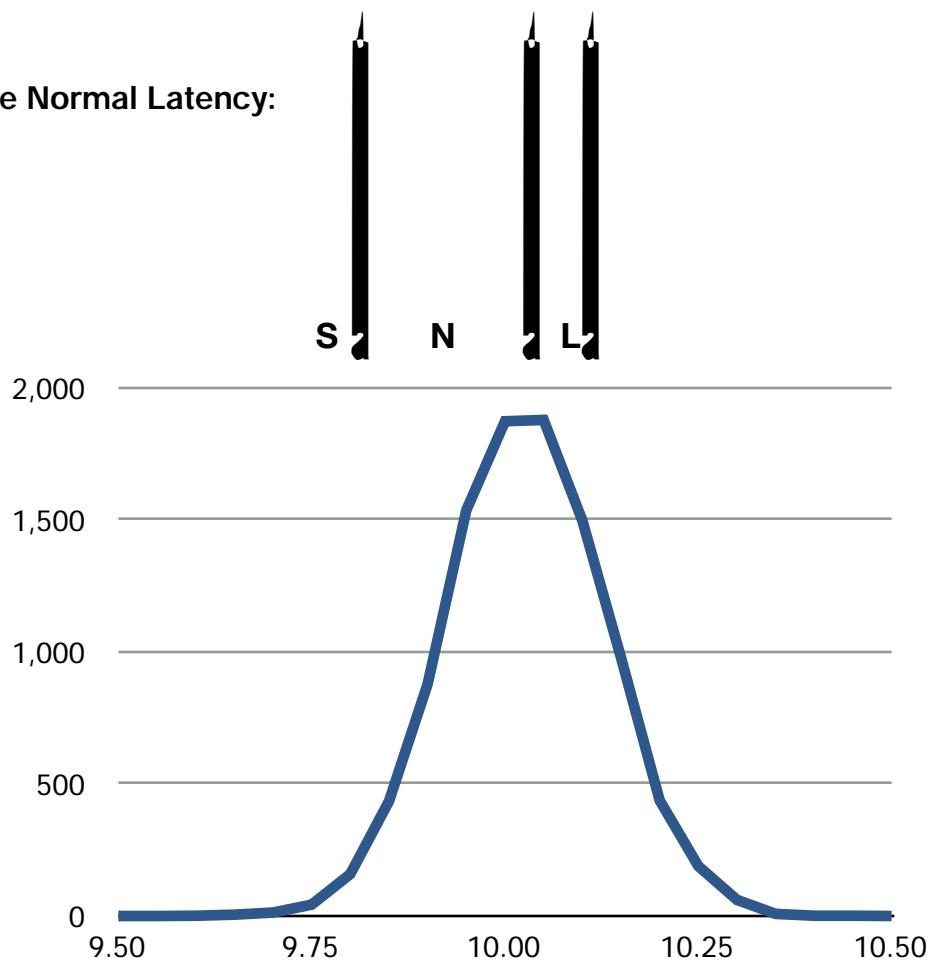
Simulation Results

IMPLEMENTATION

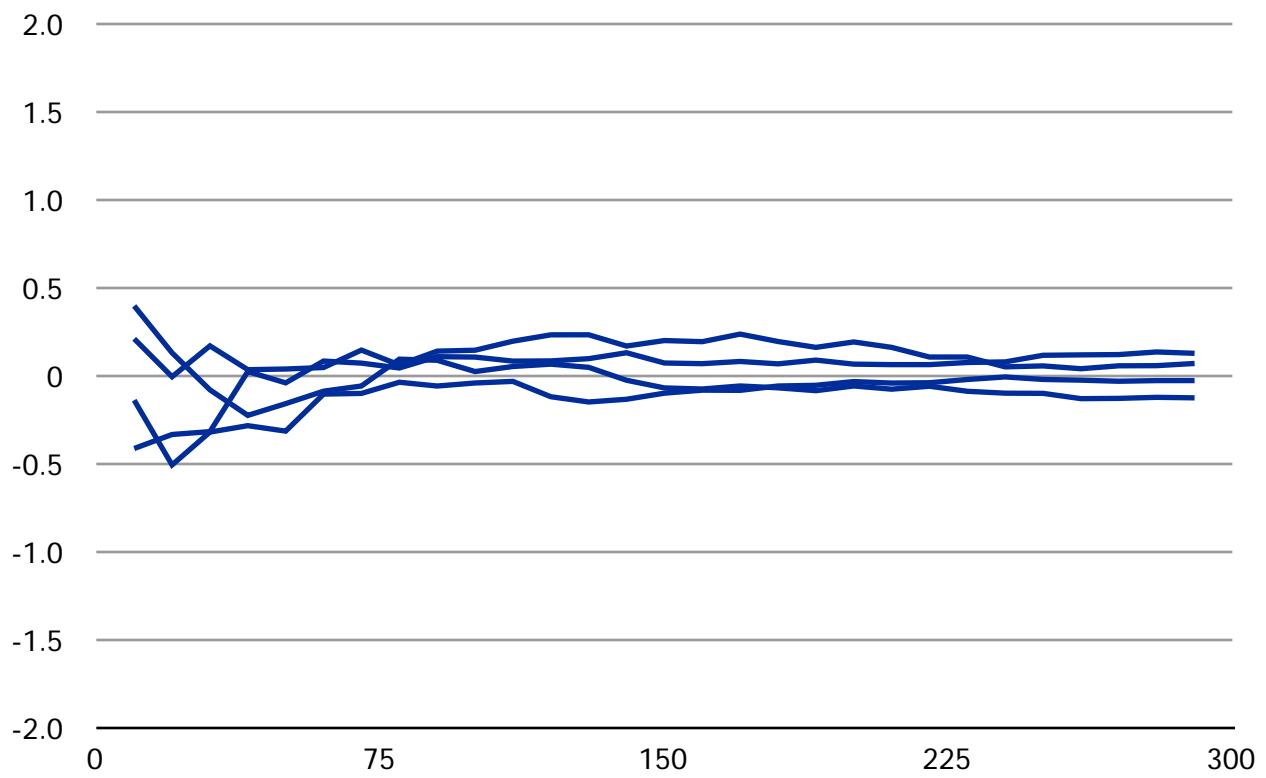
“Long-Tail” Normal Latency:

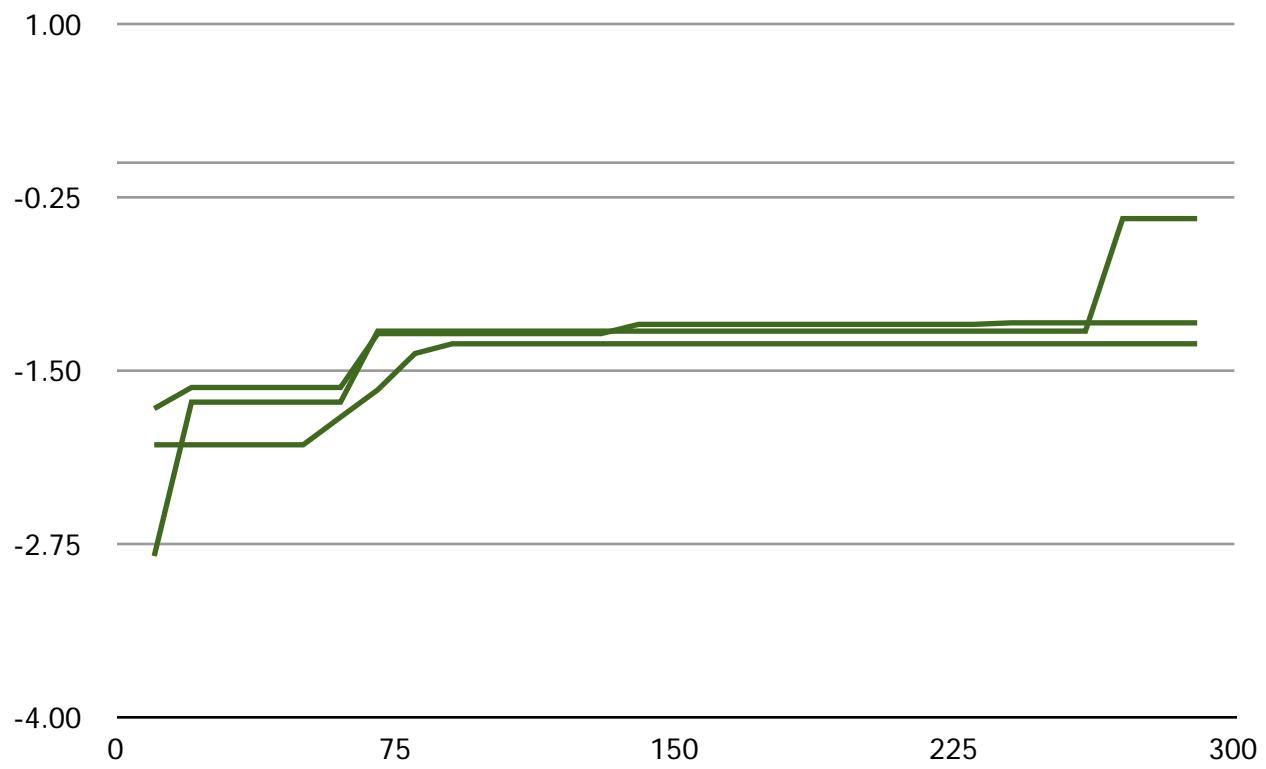


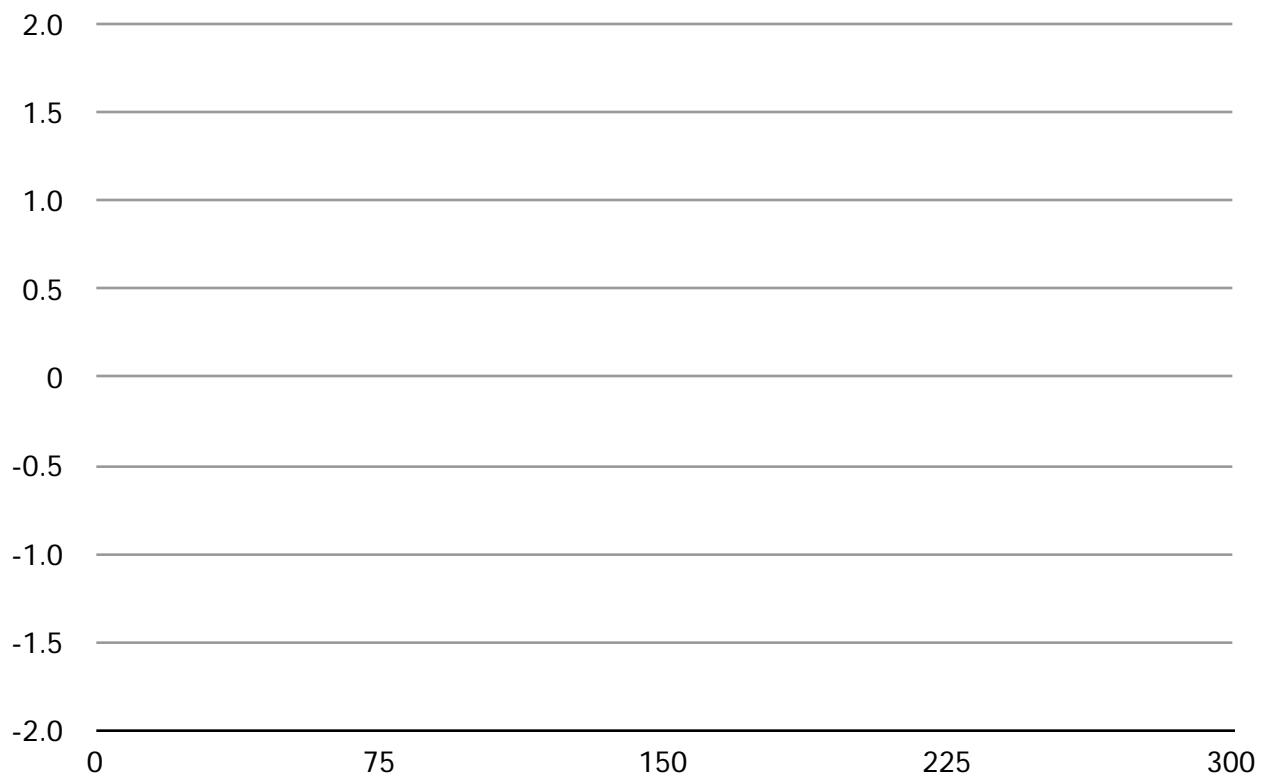
Stable Normal Latency:

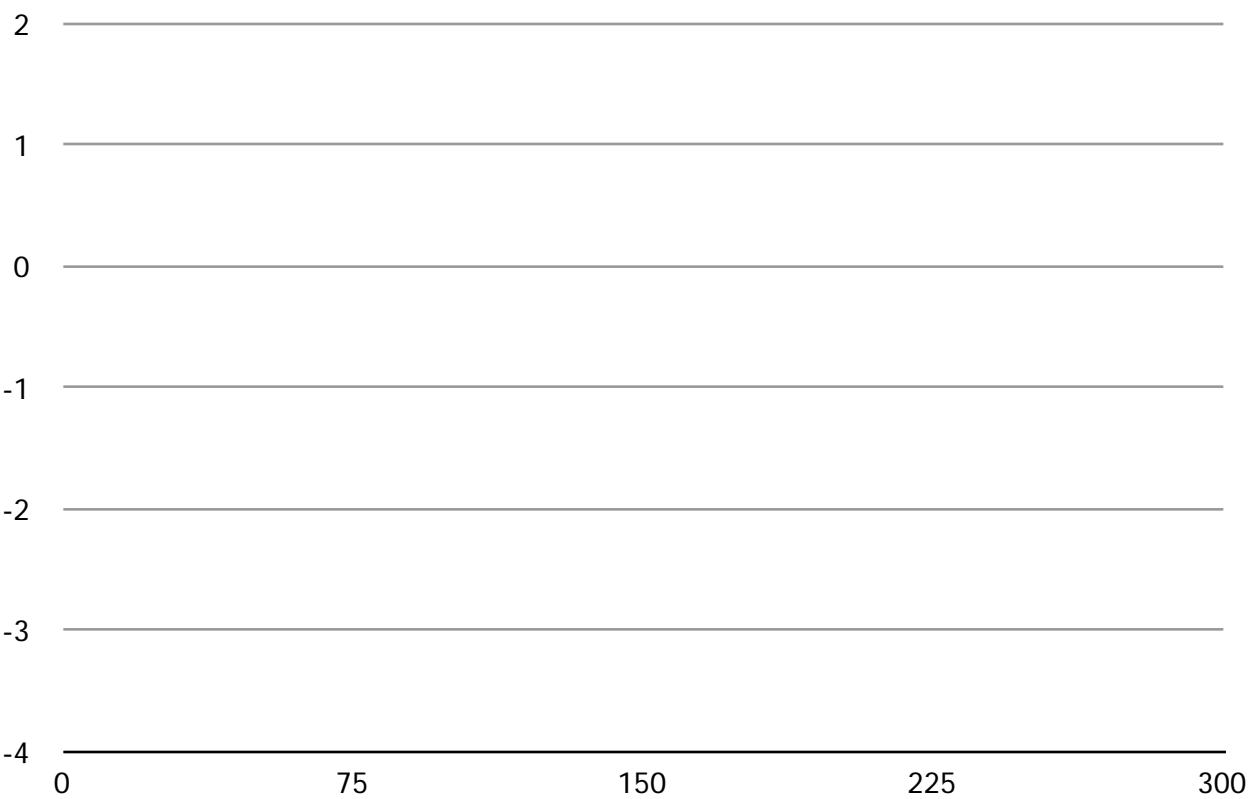


CR Min CR Mean CR Max
EF Min EF Mean EF Max
.....









Hardware Results

IMPLEMENTATION

Future Work

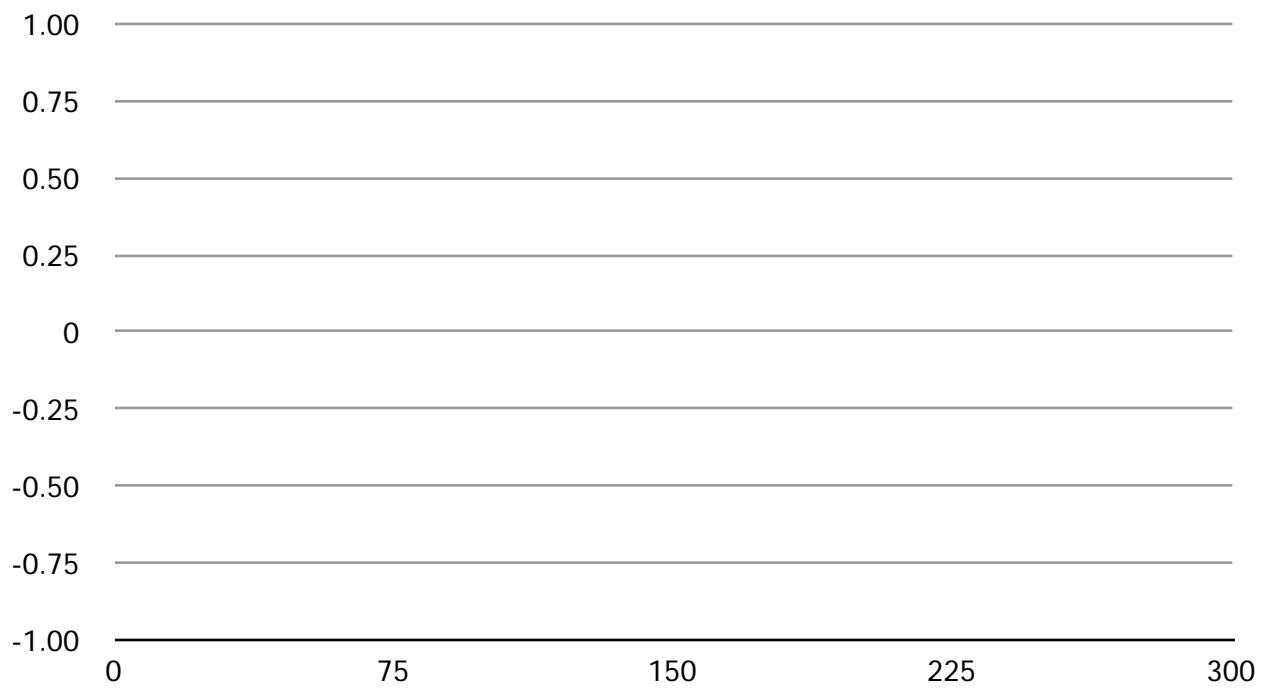
ALGORITHM IMPROVEMENTS

References

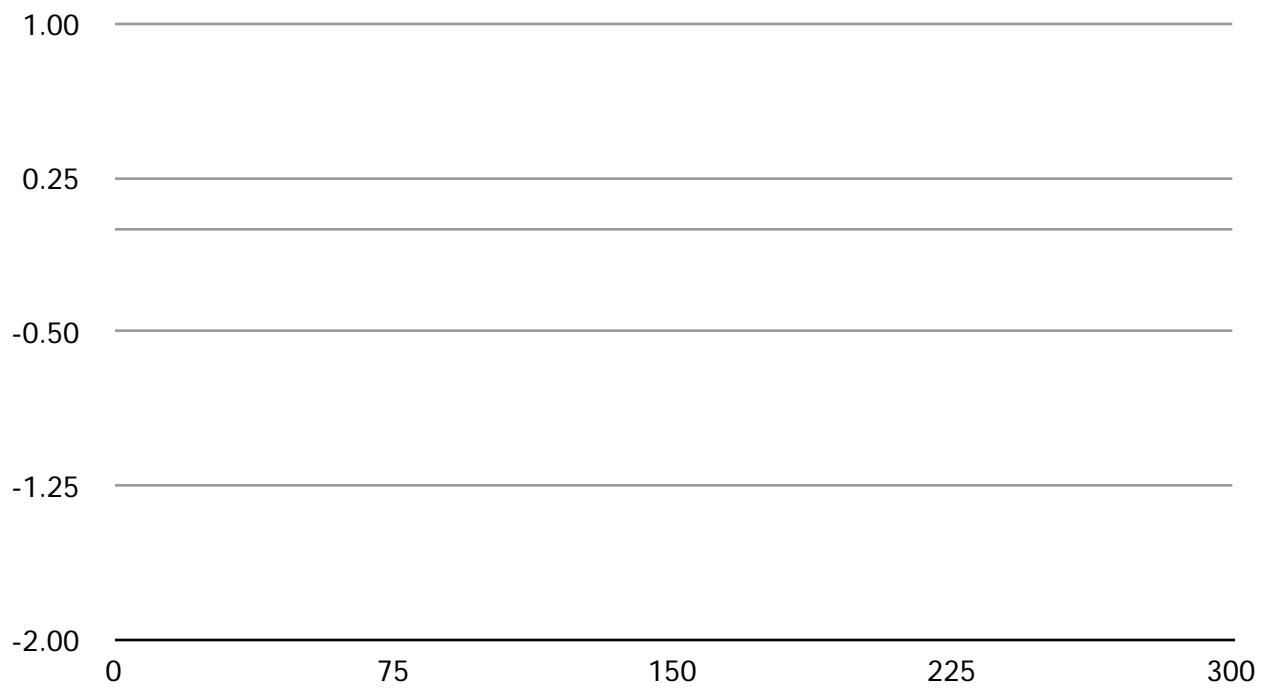
Appendix A: Simulation Results

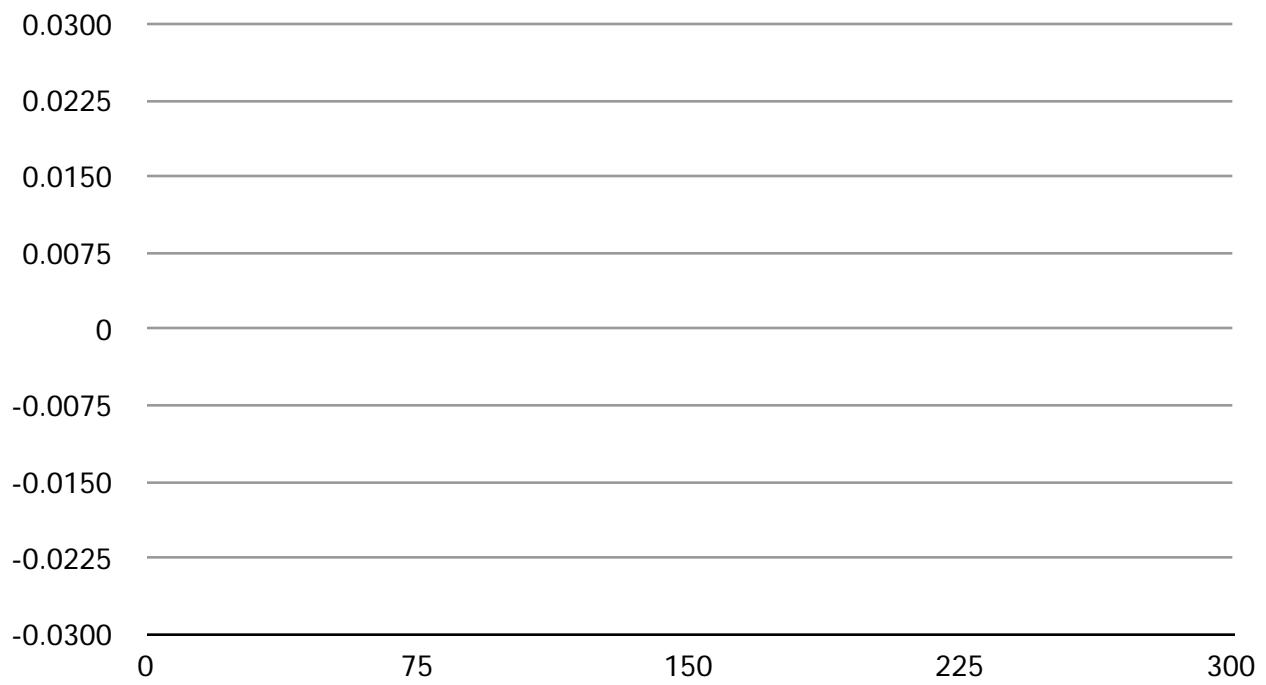
EXPERIMENT #1: BASE TIME RECORD SET

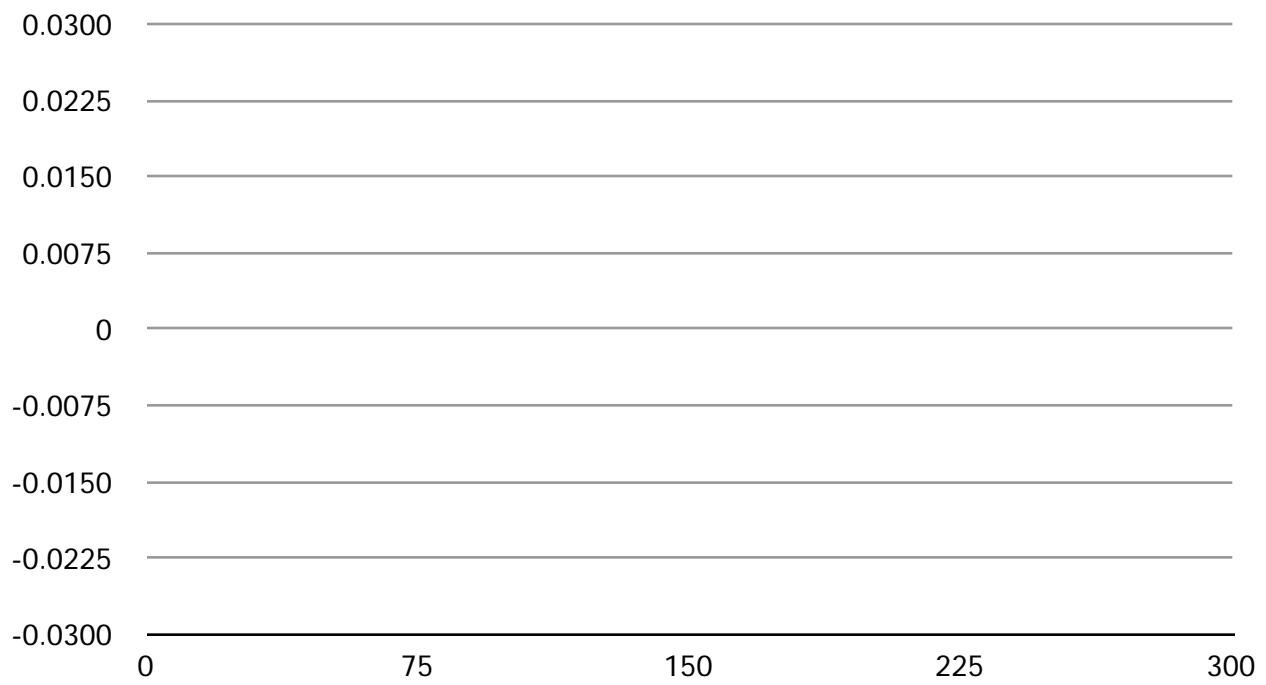
$$(\quad) + (-1) * (\quad)$$





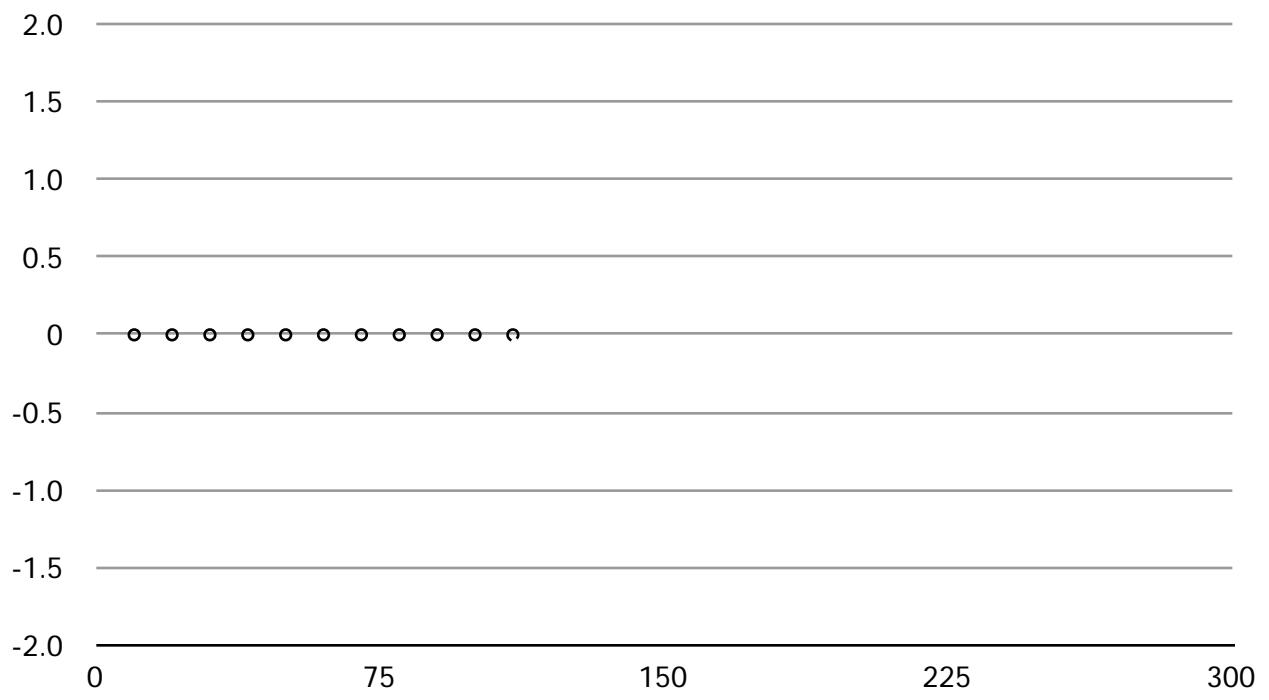


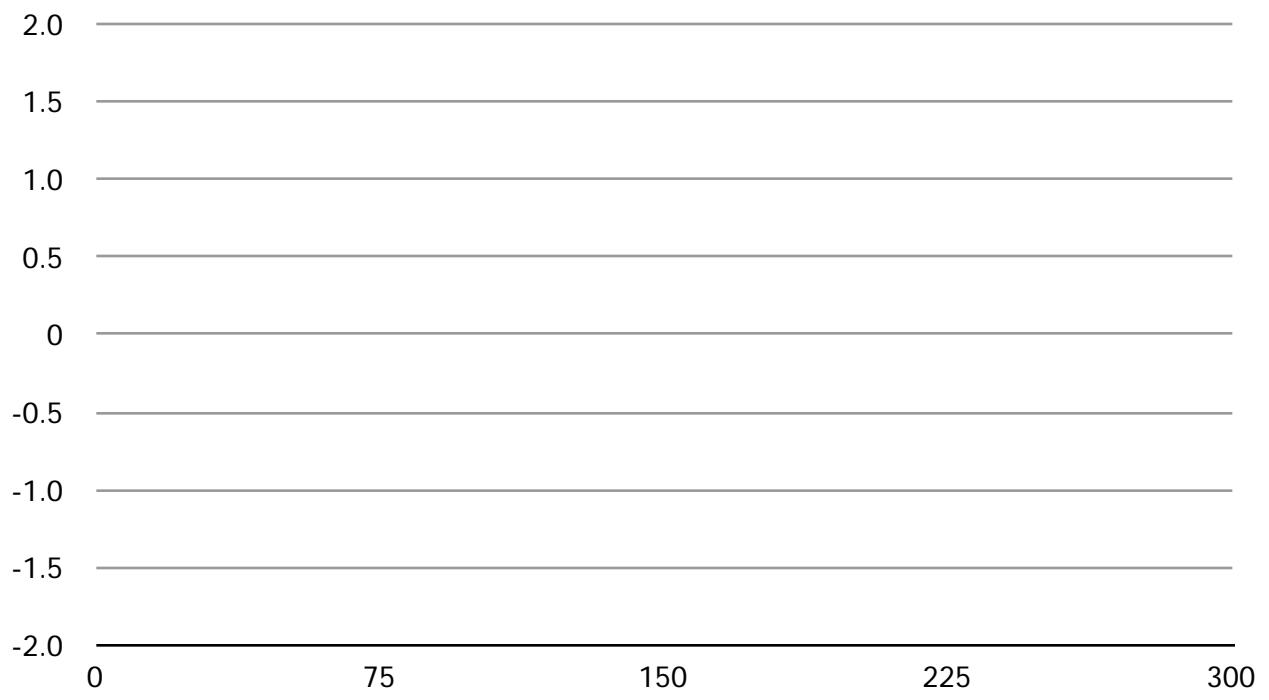




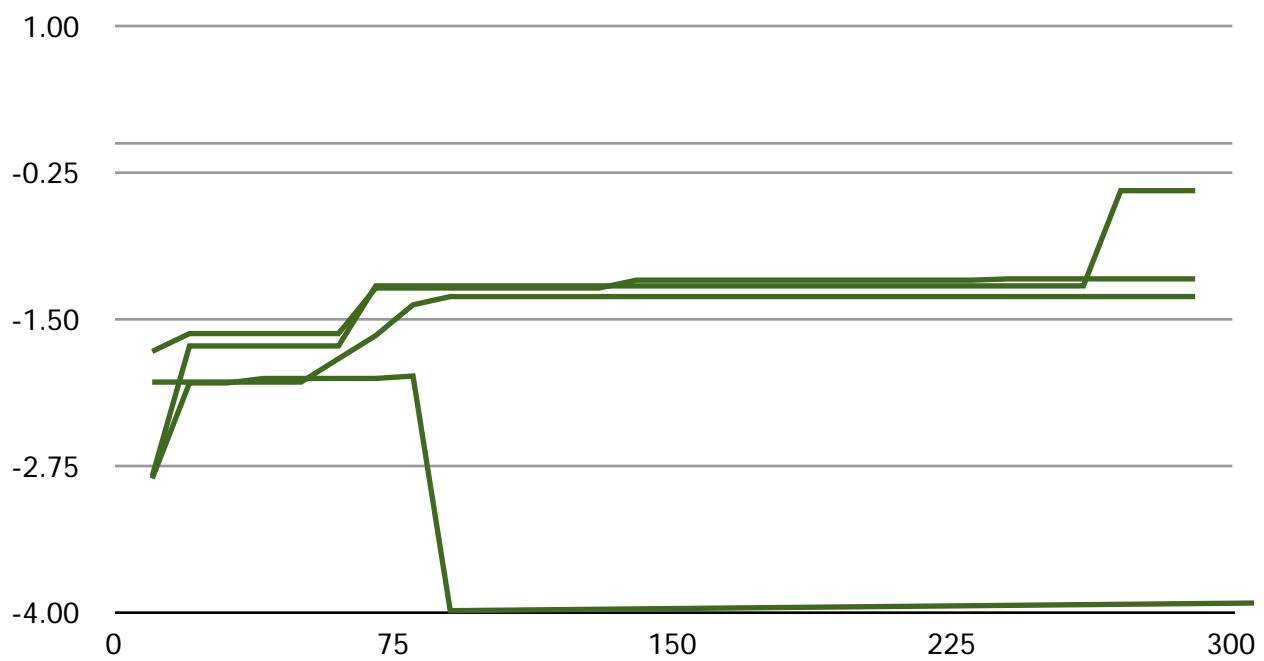
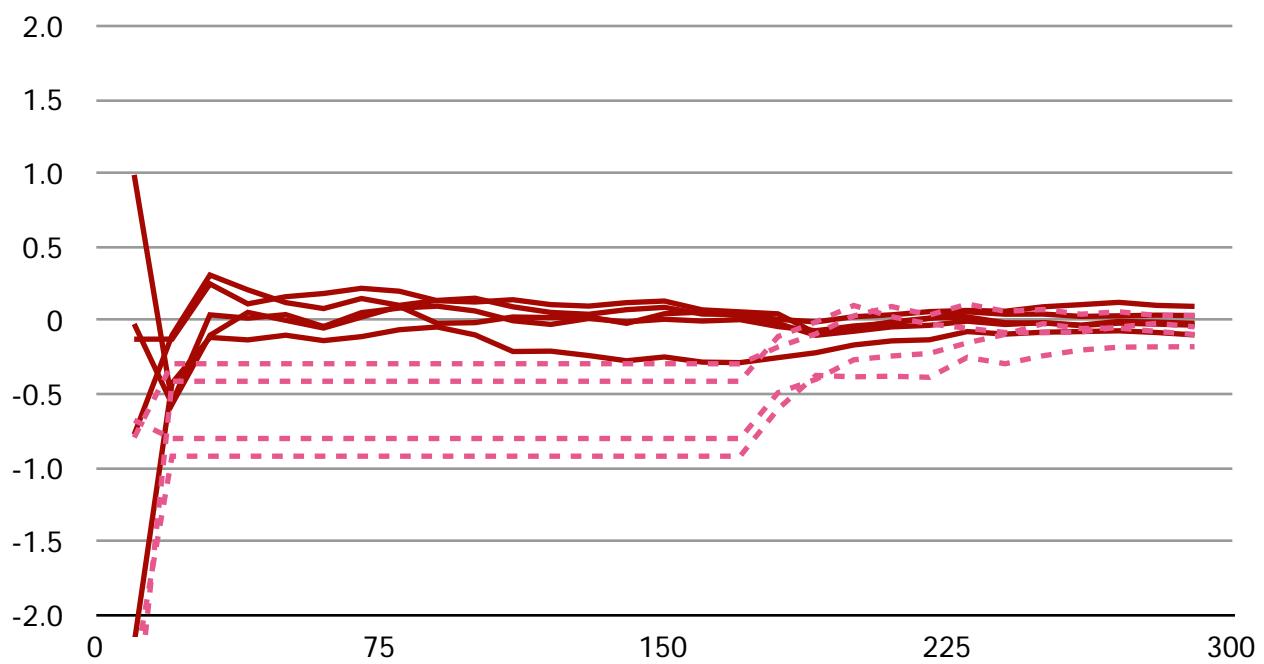
EXPERIMENT #2: CLIENT DISCONNECT

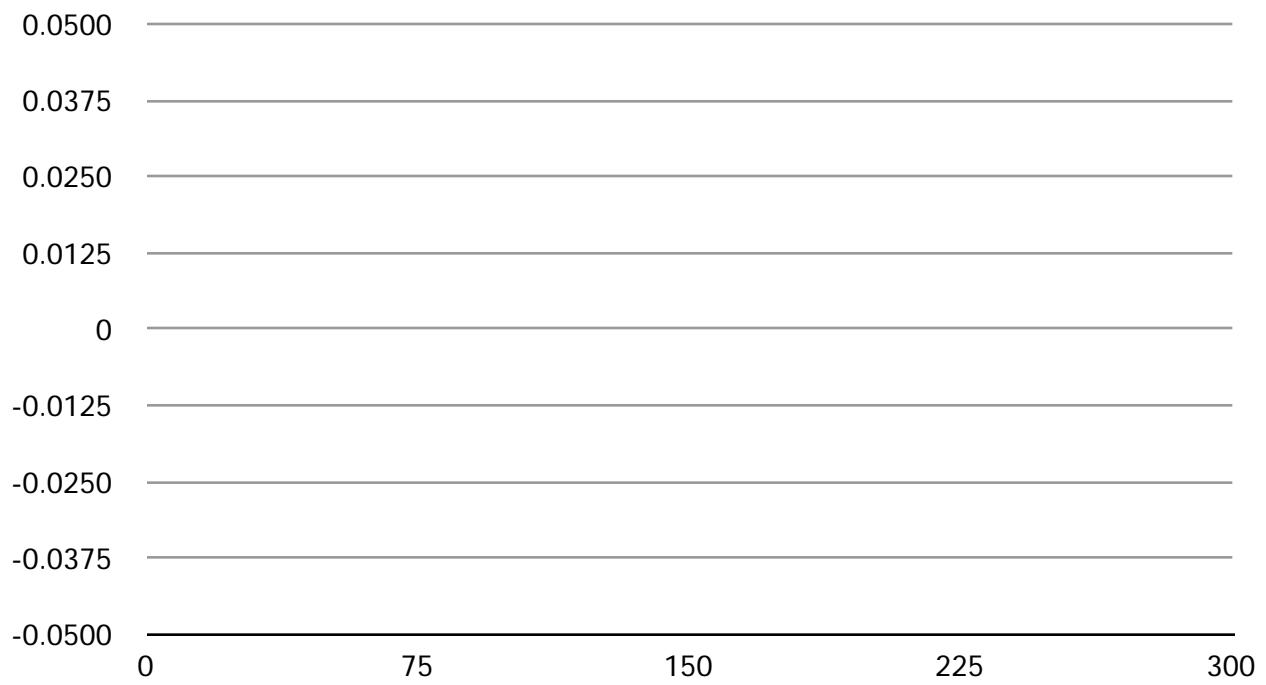
Tests:



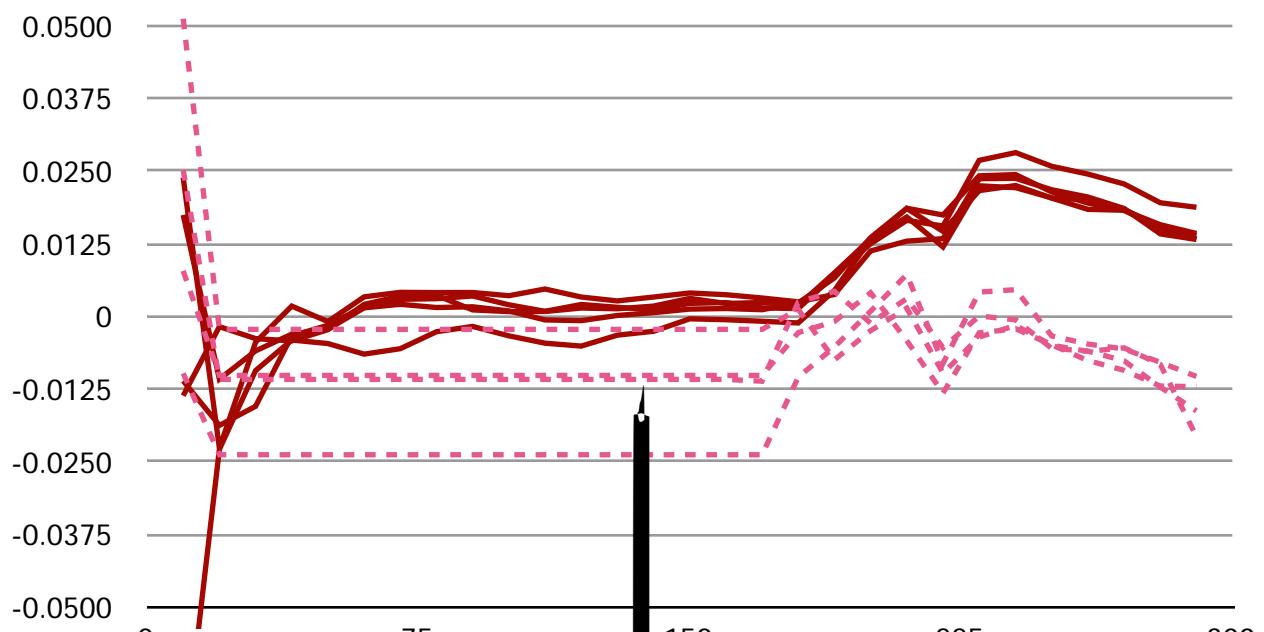


C VL-EF

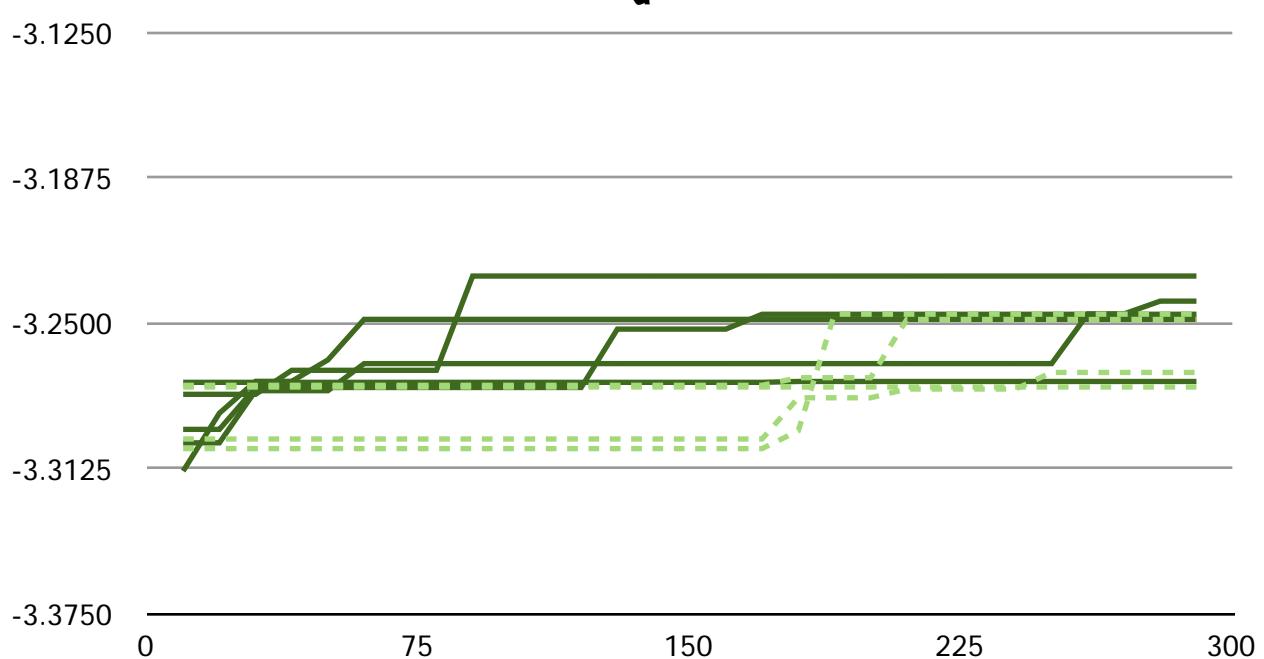




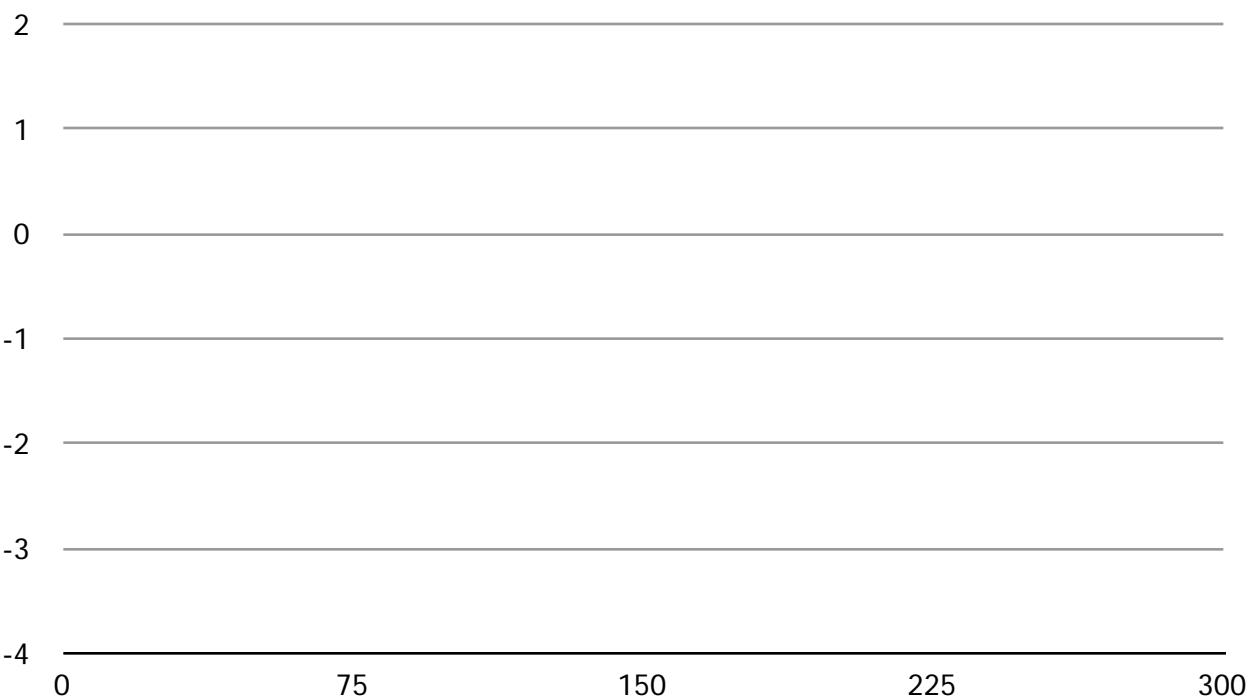
C SL-EF

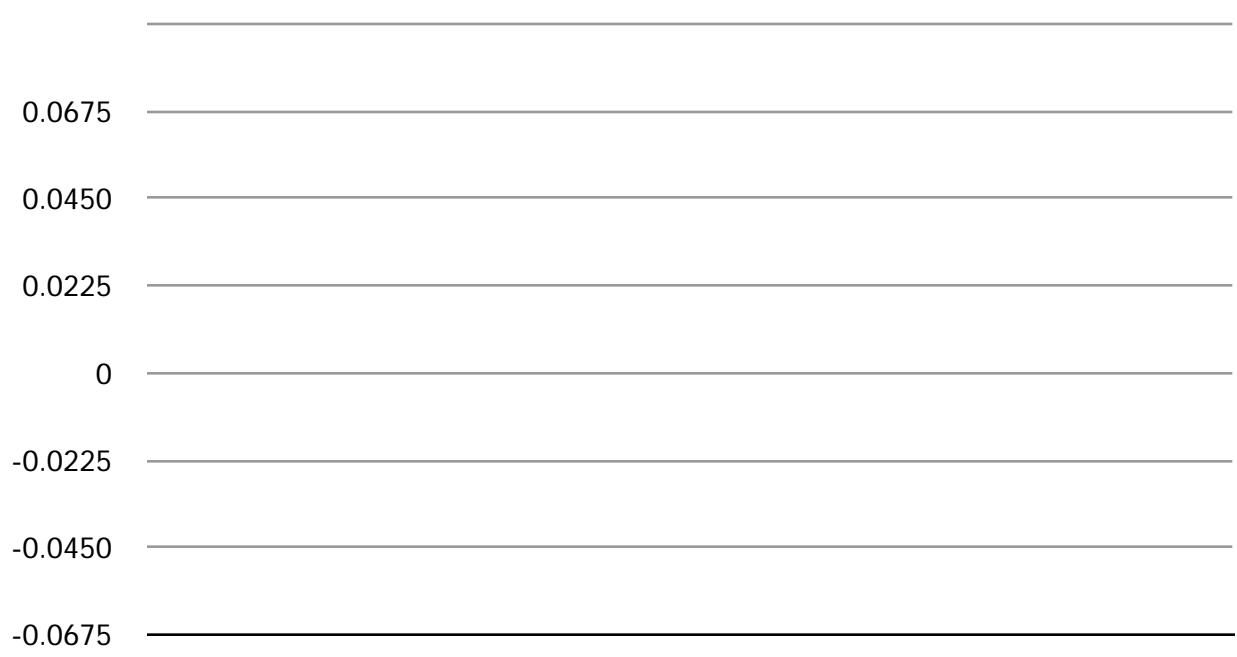


C SL-FO



EXPERIMENT #3: SERVER DISCONNECT





Appendix B: Simulation Implementation

MRFollowerNode Class

MRFollowerNode.h

```
//  
//  
//  
//  
//  
//          ( ) 2013  
//  
#      "  
#      "  
#      "  
  
,
```


(- ; ;) , *

(- , , ,)

return

#

-

()

(, , ,) * ,

- ()

(

*)

,

! (0,0)
! 0,0
! (/),
- ()
- ()
! (32 32 0 (32 ,) ++) ,
! 32 32 4 - , () + ,
!

MRTIMERecord Class

MRTIMERecord.h

```
//  
//  
//  
//  
//  
// ( ) 2013 2/21/13.  
//  
# /  
  
, ( ) * ,  
, ( ) * ,  
  
# -  
  
+ ( , , , ) , //  
+ ( , , , ) , //  
+ ( , , , ) , //  
( , , , ) * , //
```

MRTimeRecord.m

! * ,
! ! (!) .)

“

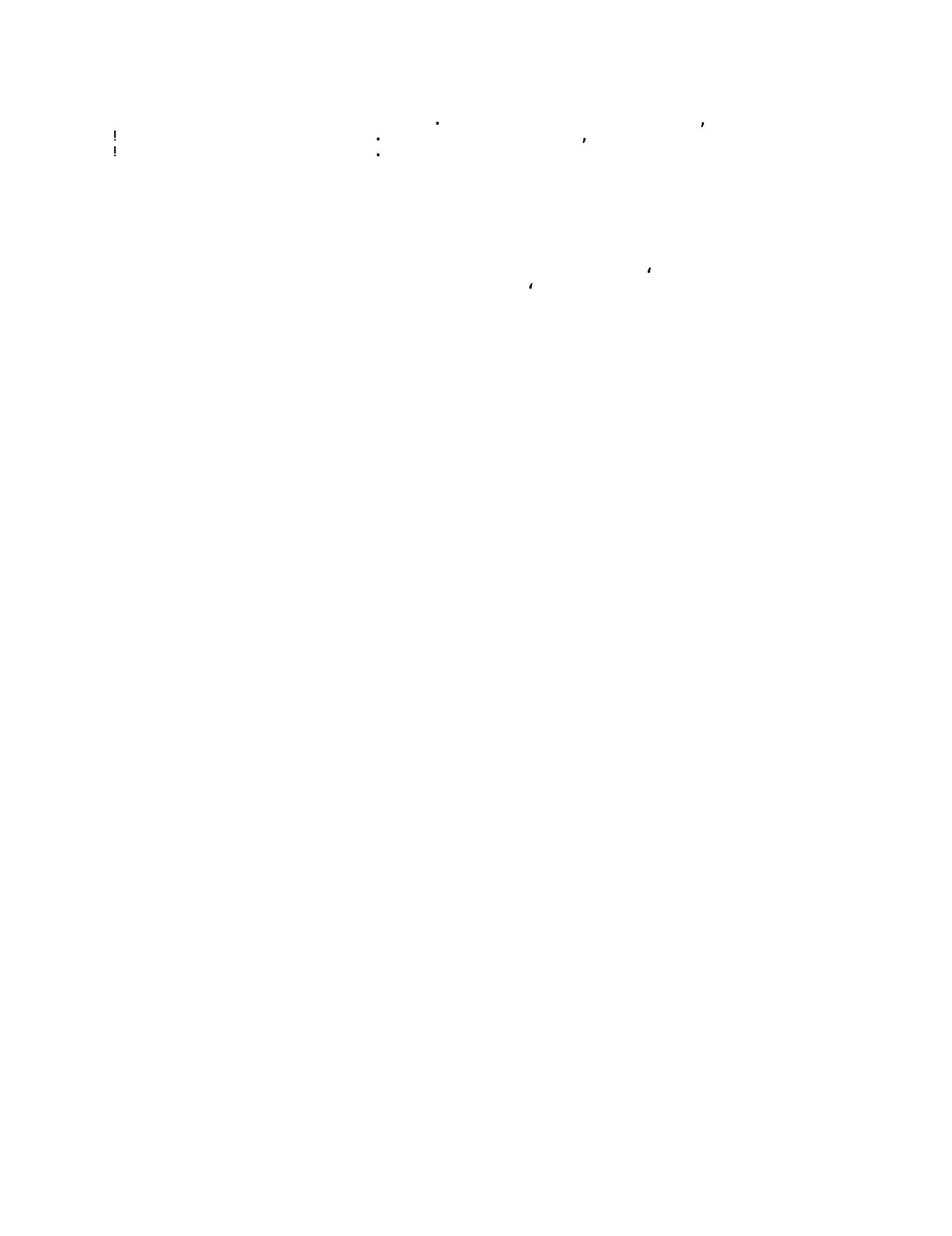
! (!),
- (*) ,
,

- () ,
! ! (-) ,
! ! 0.0 ,

- () ,
! ! (-) ,
! ! 0.0 ,

```
# -  
# ( )  
( .  
( .  
( .  
( .  
0))))  
( )  
( , , , ) * ,  
( , , , ) * ,  
( , , , ) * ,
```

-
+ (



MRTIMEUPDATEMESSAGE Class

MRTIMEUPDATEMESSAGE.h

```
//  
//  
//  
//  
//  
//          ( ) 2013           2/2 /13.  
//  
#      "  
  
      (      ;      ;      )  
      (      ;      ;      )  
+ ( )  
- ( )  
- ( )
```

MRTIMEUPDATEMESSAGE.m

```
//  
//  
//  
//  
//          ( ) 2013           2/2 /13.  
//  
#      "  
#      "  
  
      ()  
      (      ;      ;      )  
- ( )  
!      (      )  
!      !      0.0 , 0.0 ,  
!      !  
!
```

+ () () ,
!
- () () ,
!
! () , , ,
!
- () (*) , , ,
!
! , , , ,
!

MRNetworkMessage Class

MRNetworkMessage.h

```
//  
//  
//  
//  
//  
//          ( ) 2012           11/1 /12.  
//  
#           /  
  
,
```

+ ()

!

,

- ()

!

()

,

!

!

(! . , . ,
! . , . ,
(. , (. ,
! . , . ,
- (

MRCristianTimeServerNode Class

MRCristianTimeServerNode.h

```
//  
//  
//  
//  
//  
//      ( ) 2013      3/1 /13.  
//  
//
```

```
#      "  
#      "
```

```
( , , , )
```

```
! ! ( . . . && ! . . )
```

MRCristianTimeClientNode Class

MRCristianTimeClientNode.h

```
//  
//  
//  
//  
//  
//          ( ) 2013      3/1 /13.  
//  
  
#include "MRCristianTimeClientNode.h"  
  
#include <iostream>  
#include <string>  
#include <vector>  
  
using namespace std;  
  
class MRCristianTimeClientNode {  
public:  
    MRCristianTimeClientNode();  
    ~MRCristianTimeClientNode();  
  
    void connect();  
    void disconnect();  
  
    void sendTime();  
    void receiveTime();  
  
    void printTime();  
  
private:  
    string hostName;  
    int portNumber;  
    string timeString;  
};
```

MRCristianTimeClientNode.m

```
//  
//  
//  
//  
//  
//          ( ) 2013      3/1 /13.  
//  
  
#include "MRCristianTimeClientNode.h"  
  
#include <iostream>  
#include <string>  
#include <vector>  
  
using namespace std;  
  
MRCristianTimeClientNode::MRCristianTimeClientNode()  
{  
    hostName = "127.0.0.1";  
    portNumber = 12345;  
}  
  
MRCristianTimeClientNode::~MRCristianTimeClientNode()  
{  
}
```


//!

//!

0.0,

- (

! ! (-) .

- ()
! . ,
! (, ,
! !),
.

MRNetworkNode Class

MRNetworkNode.h

```
//  
//  
//  
//
```


MRNetworkDirectory Class

MRNetworkDirectory.h

MRNetworkDirectory.m

- () (*) ,

*
+ (*) ,
! ! (!) ,

,
- () ,

,
- () ,
! ,
! 10 ,
! 1.0 ,

- () (*)
! //

- () () () , 0 ,)
! (. . . ,
! (* 0.0 ,)
!

MRNetworkLatencyModel Protocol

MRNetworkLatencyModel.h

```
//  
//  
//  
//  
//  
//          11/1 /12.  
//          ( ) 2012  
//  
#  
/  
  
- ( ) ,
```

MRNormalDistrLatencyModel Class

MRNormalDistrLatencyModel.h

```
//  
//  
//  
//  
//  
// ( ) 2012 11/1 /12.  
//  
# /
```

```
{ ; ; }  
+ ( ) ( )  
( ) ( , )  
( ) ( , )  
,
```

MRNormalDistrLatencyModel.m

```
//  
//  
//  
//  
// ( ) 2012 11/1 /12.  
//  
# "
```

```
+ ( ) ( )  
( )
```

!

```
- ( ) ( )  
( )
```

```
!  
! ( )  
! !  
! ,
```

```
- ( )
```

!

1.0

0.2 ,

- (*)

! " (% , %) " ,

- ()

MRSimulation Class

MRSimulation.h

MRSimulation.m

3/10/13
 () 2013

! , () ,) * ,
- () - ,
- ,
+ () - ,
! ,
+ () (*) (,
! ,
- () ,
! , () , , ,
! ,
- () (*) , , ,
! ,
- ,
- () ,
!

• 0 ,
(" %
% %
",
),
" 2.0 ,
- ()
,,

- ()
,,

- ()
,,

! (" "),
- ()
,,

-
- ()
! ((" "),
! (" "),
! (" % "),
! (" % "),
" ,
()
! ()
! () (/)
! ! !) -2

- ()
()
! ! !
(-1),
!
- ()
*
!
(
0,0)
!
,

MRNetworkSimulation Class

MRNetworkSimulation.h

(,)

MRNetworkSimulation.m

//
//
//
//
//
//
//
//
//
" () 2013 " 3/10/13.

- () (*) ()
| | () , ,
| | , ,
| | , ,

-

- ()
! , ; ,
100 ,

- ()
! (" % " ,)

- ()
!

MRUni edAnomalySimulation Class

MRUni edAnomalySimulation.h

```
//  
//  
//  
//  
//  
//          -  
//          ( ) 2013      4/4/13.  
//  
#include "edAnomalySimulation.h"  
  
#include "edAnomaly.h"  
#include "edAnomalyList.h"  
  
#include <iostream>  
#include <vector>  
  
using namespace std;  
  
class edAnomalySimulation : public edAnomalyList  
{  
public:  
    edAnomalySimulation();  
    ~edAnomalySimulation();  
  
    void addAnomaly(edAnomaly* anomaly);  
    void removeAnomaly(edAnomaly* anomaly);  
    void printAnomalies();  
    void calculateAnomaly();  
};
```

MRUni edAnomalySimulation.m

```
//  
//  
//  
//  
//  
//          -  
//          ( ) 2013      4/4/13.  
//  
#include "edAnomalySimulation.h"  
  
#include "edAnomaly.h"  
#include "edAnomalyList.h"  
  
#include <iostream>  
#include <vector>  
  
using namespace std;  
  
class edAnomalySimulation : public edAnomalyList  
{  
public:  
    edAnomalySimulation();  
    ~edAnomalySimulation();  
  
    void addAnomaly(edAnomaly* anomaly);  
    void removeAnomaly(edAnomaly* anomaly);  
    void printAnomalies();  
    void calculateAnomaly();  
};
```


! //
! (. " " ,

“

! ! ! ! !
! ! ! ! !
+ ! ! ! !

+


```

//! !           (      1 )
//! !           (      1 ) ,
//! !           1 + , ,
//! !           ( . .
//! !           ),
//! !           ,
//! !           / 1 0 2 0 1 (     )
//! !           / 10

```

Main

main.m

*

(

Appendix C: Hardware Implementation

Main

Main.ino

/*

20 2013

* /

#

()

- (), (),

- (),

- (),
 (),
 (),
 () * ,

- (),), ,),)

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
-
()
! (00), (,),
! (,),

MRLeanTimeRecord Class

MRLeanTimeRecord.h

#

-

)

(

,

!

"

!

()

! — (),

! // (. ())
! ! . ("), ("
! !)), //

! ! (1000),

! — (),

()

// : (" ",), (),

! //

$$\begin{aligned} & \vdots & (& &) \\ & \vdots & (& , &) \\ & \vdots & (& " &), \\ & \vdots & (& 4 &) \end{aligned}$$