In [ ]:

In [ ]:

In [1]:
```
#Library Setup
from time import clock
import sys
sys.path.append("/home/thesis/ocean/organ-procurement-transplant-network/fabric/organ_chain")
from log_analysis import PeerContainer, get_time, time_to_commit, set_blocks_info
#Graphs
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

peer_names=["peer0.gp.organ.com",
           "peer1.gp.organ.com",
            "peer0.histocompatibility.organ.com",
            "peer1.histocompatibility.organ.com",
            "peer0.opo.organ.com",
            "peer1.opo.organ.com",
            "peer0.hospital.organ.com",
            "peer1.hospital.organ.com"]

base = "/home/thesis/ocean/organ-procurement-transplant-network/fabric/organ_chain/TOPO2/"

class TxnTimes:
    'This is txn times'
    def __init__(self, receive_time = 0, validate_time = 0, commmit_time = 0):
        self.receive_time = receive_time
        self.validate_time = validate_time
        self.commmit_time = commmit_time

class Block:
    def __init__(self, block_number):
        self.__name__ = str(block_number)
        self.block_number = block_number
        self.peers_times = {}

    def add_peer(self, peer_name, block_receive_time, block_validated_time, block_commited_time, verbose=False):
        '''
            This involves getting the block receive time, block validated time and the block commit time.
        '''

        t = TxnTimes(block_receive_time, block_validated_time, block_commited_time)

        self.peers_times[peer_name] = t
        if(verbose is True):
            print('Adding %s with rt %s, vt %s and ct %s' %(peer_name, block_receive_time, block_validated_time, block_commited_time))

    def get_longest_commit_time():
        largest = self.peer.itervalues().next()[2]
        for peer_id, time_array in self.peer:
```

```
            if(time_array[2]):
                pass
    def print_block_info():
        print("Block Number %s" % self.block_number)
        print("Peer Informa")

    def get_smallest_commit_time():
        pass

    def get_propogation_time(self, peers):
        commit_times = []
        receive_times = []
        for peer in self.peers_times:
            commit_times.append(self.peers_times[peer].commmit_time)
            receive_times.append(self.peers_times[peer].receive_time)
        max_commit_time = max(commit_times)
        min_receive_time = min(receive_times)
        return(max_commit_time - min_receive_time)

def get_propogation_times(log_files):
    '''
        STEPS:
            1. Setup the all_peers {}
            2. Extracct receive times
            3. Extract validation times
            4. Extract commit times
            5. Return Propogation times
    '''
    print("Starting STEP 1: Setup the all_peers dict")
    all_peers = {}
    for container in peer_names:

        start = clock()
        all_peers[container] = PeerContainer(str(container), log_file
s[container])
        set_blocks_info(all_peers[container])
        delta = clock() - start
        print("\t Set logs for %s in %s Sec" % (container, delta))

    print("Starting Step 2: Extract receive times.")

    # Adding block received times to the PeerContainer object
    start = 0
    start = clock()
    for peer in peer_names:
        l = ((all_peers[peer].find_received_block()))
#         print('\tThere are %s Recevied Lines in %s' % (len(l), pee
r))
        number_of_blocks = len(l)
        for line in l:
            pattern = 'Received block ['
            start = line['log'].find(pattern)
            if start == -1:
                continue
            x = line['log'][start + len(pattern):]
            end = x.find(']')
            if end == -1:
```

```
                continue
            block_number = x[:end]
            time = get_time(line)
            all_peers[peer].block_times[int(block_number)] = [time]
    delta = clock() - start
    print("\t Done Extracting Receive Times in %s Sec" % (delta))
    print("Stating Step 3: Extract validation times.")
    start = 0
    start = clock()
    for peer in peer_names:
        l = ((all_peers[peer].find_validated_block()))
#         print('\tThere are %s Validated Lines in %s' % (len(l), pee
r))
        number_of_blocks = len(l)
        for line in l:
            pattern = 'Validated block ['
            start = line['log'].find(pattern)
            if start == -1:
                continue
            x = line['log'][start + len(pattern):]
            end = x.find(']')
            if end == -1:
                continue
            block_number = x[:end]
            time = get_time(line)
            try:
                all_peers[peer].block_times[int(block_number)].append
(time)
            except:
                pass
    #         print('%s block was Validated at %s by peer %s' % (bloc
k_number, time, all_peers[peer].name))
    delta = clock() - start
    print("\t Done Extracting Validation Times in %s Sec" % (delta))
    print("Stating Step 4: Extract commit times.")
    start = 0
    start = clock()
    for peer in peer_names:
        l = ((all_peers[peer].find_commited_block()))
#         print('\tThere are %s Committed Line %s' % (len(l), peer))
        number_of_blocks = len(l)
        for line in l:
            pattern = 'Committed block ['
            start = line['log'].find(pattern)
            if start == -1:
                continue
            x = line['log'][start + len(pattern):]
            end = x.find(']')
            if end == -1:
                continue
            block_number = x[:end]
            time = get_time(line)
            try:
                all_peers[peer].block_times[int(block_number)].append
(time)
            except:
                pass
```

```
    delta = clock() - start
    print("\t Done Extracting Validation Times in %s Sec" % (delta))
    print("Starting Step 5: Return Propogation Time")
    blocks = {}
    # print(all_peers)
    for j, peer in enumerate(all_peers):
        print("\t Processing %s" % peer)
    #     print(all_peers[peer].block_times)
    #     print(len(all_peers[peer].block_times))
        for i in range(1, len(all_peers[peer].block_times)):
            rt = all_peers[peer].block_times[i][0]
            vt = all_peers[peer].block_times[i][1]
            ct = all_peers[peer].block_times[i][2]
    #         print("rt %s; vt %s; ct %s"% (rt, vt, ct))
            if i not in blocks:
                blocks[i] = Block(i)
            blocks[i].add_peer(all_peers[peer].name, rt, vt, ct)
#         print(all_peers[peer].name)

    propogation_times = []
    for i in range(1, len(blocks)):
#         print('Propogation Time of %s is %s' % (i, blocks[i].get_pr
opogation_time(all_peers[peer])))
        propogation_times.append(blocks[i].get_propogation_time(all_p
eers[peer]).total_seconds())
    print("Done")
    return(propogation_times)
```

Experiment 1

In [6]:
```
experiment1Results = []
exp1dir = "/home/thesis/ocean/organ-procurement-transplant-network/fa
bric/organ_chain/TOPO2/Experiment1(BatchTImeout)/logs_1_"
```

In [7]:
```
#Experiment 1 Effect of Time Out Duration
log_files = {}

for container in peer_names:
    log_files[container] = ('%s1/%s.log'%(exp1dir,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment1Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.025616 Sec
         Set logs for peer1.gp.organ.com in 0.026366 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.026123
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.02664 S
ec
         Set logs for peer0.opo.organ.com in 0.029892 Sec
         Set logs for peer1.opo.organ.com in 0.023251 Sec
         Set logs for peer0.hospital.organ.com in 0.030989 Sec
         Set logs for peer1.hospital.organ.com in 0.027428 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -90.757196 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -94.096549 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -89.477814 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

In [8]:
```
#Experiment 1.1 Batch Time Out of 5 sec with transaction rate of
log_files = {}
for container in peer_names:
    log_files[container] = ('%s2/%s.log'%(exp1dir,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment1Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict

---------------------------------------------------------------------
------
IOError                                   Traceback (most recent call
last)
<ipython-input-8-4c4545db9069> in <module>()
      4     log_files[container] = ('%s2/%s.log'%(exp1dir,container))
      5 # print log_files
----> 6 propogation_time = get_propogation_times(log_files)
      7 experiment1Results.append(propogation_time)

<ipython-input-1-5406aa3f7b73> in get_propogation_times(log_files)
     80
     81         start = clock()
---> 82         all_peers[container] = PeerContainer(str(container),
 log_files[container])
     83         set_blocks_info(all_peers[container])
     84         delta = clock() - start

/home/thesis/ocean/organ-procurement-transplant-network/fabric/organ_
chain/log_analysis.pyc in __init__(self, name, log_file)
     19         self.name = name
     20         self.log_file = []
---> 21         with open(log_file, 'r') as f:
     22             for line in f:
     23                 self.log_file.append(json.loads(line))

IOError: [Errno 2] No such file or directory: '/home/thesis/ocean/org
an-procurement-transplant-network/fabric/organ_chain/TOPO2/Experiment
1(BatchTImeout)/logs_1_2/peer0.gp.organ.com.log'
```

In [9]:
```
#Experiment 1.3 Batch Time Out of 5 sec with transaction rate of
log_files = {}
for container in peer_names:
    log_files[container] = ('%s3/%s.log'%(exp1dir,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment1Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.019631 Sec
         Set logs for peer1.gp.organ.com in 0.024502 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.022286
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.023786
Sec
         Set logs for peer0.opo.organ.com in 0.023845 Sec
         Set logs for peer1.opo.organ.com in 0.024262 Sec
         Set logs for peer0.hospital.organ.com in 0.025089 Sec
         Set logs for peer1.hospital.organ.com in 0.027816 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -88.307603 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -91.626359 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -86.936145 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

In [10]:
```
#Experiment 1.4 Batch Time Out of 5 sec with transaction rate of
log_files = {}
for container in peer_names:
    log_files[container] = ('%s4/%s.log'%(exp1dir,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment1Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.024939 Sec
         Set logs for peer1.gp.organ.com in 0.024125 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.022977
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.024615
Sec
         Set logs for peer0.opo.organ.com in 0.023774 Sec
         Set logs for peer1.opo.organ.com in 0.023167 Sec
         Set logs for peer0.hospital.organ.com in 0.024409 Sec
         Set logs for peer1.hospital.organ.com in 0.026409 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -86.028637 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -89.354367 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -84.661637 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

Experiment 2

Experiment 2 Steps

1. Read the log files for each peer
2. Extract Lines 2.1 Received Lines 2.2 Validated Lines 2.3 Commited Lines
3. Find the propogation time
4. Make graphs

In [30]:
```
experiment2Results =[]
exp1dir = "/home/thesis/ocean/organ-procurement-transplant-network/fa
bric/organ_chain/TOPO2/Experiment1(BatchTImeout)/logs_1_"
```

In [31]:
```
#Experiment 2.1
log_files = {}
base = "/home/thesis/ocean/organ-procurement-transplant-network/fabri
c/organ_chain/TOPO2/Experiment2(PreferredMaxBytes)/logs_2_"
for container in peer_names:
    log_files[container] = ('%s1/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment2Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
        Set logs for peer0.gp.organ.com in 0.027964 Sec
        Set logs for peer1.gp.organ.com in 0.027046 Sec
        Set logs for peer0.histocompatibility.organ.com in 0.021924
Sec
        Set logs for peer1.histocompatibility.organ.com in 0.024407
Sec
        Set logs for peer0.opo.organ.com in 0.025243 Sec
        Set logs for peer1.opo.organ.com in 0.030736 Sec
        Set logs for peer0.hospital.organ.com in 0.027363 Sec
        Set logs for peer1.hospital.organ.com in 0.026696 Sec
Starting Step 2: Extract receive times.
        Done Extracting Receive Times in 25.331384 Sec
Stating Step 3: Extract validation times.
        Done Extracting Validation Times in 22.119528 Sec
Stating Step 4: Extract commit times.
        Done Extracting Validation Times in 26.925725 Sec
Starting Step 5: Return Propogation Time
        Processing peer0.histocompatibility.organ.com
        Processing peer1.histocompatibility.organ.com
        Processing peer1.opo.organ.com
        Processing peer0.gp.organ.com
        Processing peer0.hospital.organ.com
        Processing peer0.opo.organ.com
        Processing peer1.hospital.organ.com
        Processing peer1.gp.organ.com
Done
```

In [32]:
```
#Experiment 2.2
log_files = {}

for container in peer_names:
    log_files[container] = ('%s2/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment2Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.033774 Sec
         Set logs for peer1.gp.organ.com in 0.022714 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.026383
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.023517
Sec
         Set logs for peer0.opo.organ.com in 0.025848 Sec
         Set logs for peer1.opo.organ.com in 0.023241 Sec
         Set logs for peer0.hospital.organ.com in 0.027447 Sec
         Set logs for peer1.hospital.organ.com in 0.028698 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in 28.090339 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in 26.829375 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in 35.33935 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

In [33]:
```
#Experiment 2.3
log_files = {}
for container in peer_names:
    log_files[container] = ('%s3/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment2Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.195748 Sec
         Set logs for peer1.gp.organ.com in 0.184146 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.186521
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.178483
Sec
         Set logs for peer0.opo.organ.com in 0.180438 Sec
         Set logs for peer1.opo.organ.com in 0.186993 Sec
         Set logs for peer0.hospital.organ.com in 0.190793 Sec
         Set logs for peer1.hospital.organ.com in 0.197103 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in 41.317123 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in 41.514565 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in 50.047283 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

In [34]:
```
#Experiment 2.4
log_files = {}

for container in peer_names:
    log_files[container] = ('%s4/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
experiment2Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.20569 Sec
         Set logs for peer1.gp.organ.com in 0.174978 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.182616
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.175668
Sec
         Set logs for peer0.opo.organ.com in 0.165815 Sec
         Set logs for peer1.opo.organ.com in 0.176333 Sec
         Set logs for peer0.hospital.organ.com in 0.586374 Sec
         Set logs for peer1.hospital.organ.com in 0.164306 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in 57.433175 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in 57.768651 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in 62.877174 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
```

Experiment 3

In [15]:
```
experiment3Results =[]
base = "/home/thesis/ocean/organ-procurement-transplant-network/fabri
c/organ_chain/TOPO2/Experiment3(EndorsementPolicy)/logs_3_"
```

In [16]:
```
#Experiment 3.1
log_files = {}
for container in peer_names:
    log_files[container] = ('%s1/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment3Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.022769 Sec
         Set logs for peer1.gp.organ.com in 0.028869 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.02373 S
ec
         Set logs for peer1.histocompatibility.organ.com in 0.025538
Sec
         Set logs for peer0.opo.organ.com in 0.025198 Sec
         Set logs for peer1.opo.organ.com in 0.025108 Sec
         Set logs for peer0.hospital.organ.com in 0.028783 Sec
         Set logs for peer1.hospital.organ.com in 0.027306 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -62.521021 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -65.725631 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -60.895485 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [17]:
```
#Experiment 3.2
log_files = {}
for container in peer_names:
    log_files[container] = ('%s2/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment3Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.03051 Sec
         Set logs for peer1.gp.organ.com in 0.026953 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.029944
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.028152
Sec
         Set logs for peer0.opo.organ.com in 0.023076 Sec
         Set logs for peer1.opo.organ.com in 0.022825 Sec
         Set logs for peer0.hospital.organ.com in 0.027246 Sec
         Set logs for peer1.hospital.organ.com in 0.031776 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -59.931048 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -63.203079 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -58.452923 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [18]:
```
#Experiment 3.3
log_files = {}

for container in peer_names:
    log_files[container] = ('%s3/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment3Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.15622 Sec
         Set logs for peer1.gp.organ.com in 0.158765 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.155405
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.158555
Sec
         Set logs for peer0.opo.organ.com in 0.167119 Sec
         Set logs for peer1.opo.organ.com in 0.160741 Sec
         Set logs for peer0.hospital.organ.com in 0.160144 Sec
         Set logs for peer1.hospital.organ.com in 0.157113 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -51.954214 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -51.329789 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -42.651839 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [19]:
```
#Experiment 3.4
log_files = {}

for container in peer_names:
    log_files[container] = ('%s4/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment3Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.133571 Sec
         Set logs for peer1.gp.organ.com in 0.104261 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.108325
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.059112
Sec
         Set logs for peer0.opo.organ.com in 0.073193 Sec
         Set logs for peer1.opo.organ.com in 0.078876 Sec
         Set logs for peer0.hospital.organ.com in 0.050784 Sec
         Set logs for peer1.hospital.organ.com in 0.047801 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -40.589286 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -43.220248 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -37.87821 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

Experiment 4

In [20]:
```
experiment4Results = []
base = "/home/thesis/ocean/organ-procurement-transplant-network/fabri
c/organ_chain/TOPO2/Experiment4(TransactionRate)/logs_4_"
```

In [21]:
```
#Experiment 4.1
log_files = {}

for container in peer_names:
    log_files[container] = ('%s1/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment4Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.174822 Sec
         Set logs for peer1.gp.organ.com in 0.166658 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.145895
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.162173
Sec
         Set logs for peer0.opo.organ.com in 0.1913 Sec
         Set logs for peer1.opo.organ.com in 0.163138 Sec
         Set logs for peer0.hospital.organ.com in 0.186641 Sec
         Set logs for peer1.hospital.organ.com in 0.172865 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -31.932465 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -34.730349 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -29.820489 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [22]:
```
#Experiment 4.2
log_files = {}

for container in peer_names:
    log_files[container] = ('%s2/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment4Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.025216 Sec
         Set logs for peer1.gp.organ.com in 0.023598 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.025049
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.023736
Sec
         Set logs for peer0.opo.organ.com in 0.02607 Sec
         Set logs for peer1.opo.organ.com in 0.02481 Sec
         Set logs for peer0.hospital.organ.com in 0.023874 Sec
         Set logs for peer1.hospital.organ.com in 0.028137 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -28.694751 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -31.894224 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -27.085688 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [23]:
```
#Experiment 4.3
log_files = {}

for container in peer_names:
    log_files[container] = ('%s3/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment4Results.append(propogation_time)
```

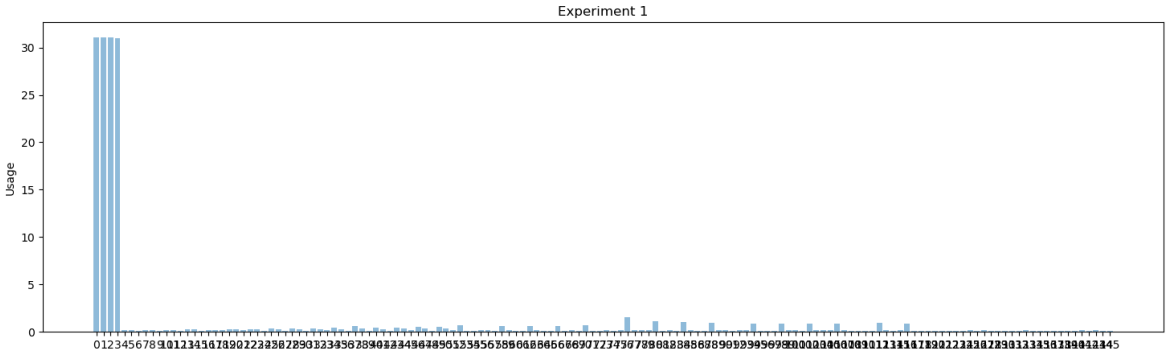
```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.031587 Sec
         Set logs for peer1.gp.organ.com in 0.028118 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.027139
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.026874
Sec
         Set logs for peer0.opo.organ.com in 0.028841 Sec
         Set logs for peer1.opo.organ.com in 0.024579 Sec
         Set logs for peer0.hospital.organ.com in 0.028667 Sec
         Set logs for peer1.hospital.organ.com in 0.028027 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -26.065219 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -29.258826 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -24.357637 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

In [24]:
```
#Experiment 4.4
log_files = {}

for container in peer_names:
    log_files[container] = ('%s4/%s.log'%(base,container))
# print log_files
propogation_time = get_propogation_times(log_files)
print("The length of propogation time is %s" % len(propogation_time))
experiment4Results.append(propogation_time)
```

```
Starting STEP 1: Setup the all_peers dict
         Set logs for peer0.gp.organ.com in 0.150447 Sec
         Set logs for peer1.gp.organ.com in 0.188971 Sec
         Set logs for peer0.histocompatibility.organ.com in 0.160562
Sec
         Set logs for peer1.histocompatibility.organ.com in 0.164328
Sec
         Set logs for peer0.opo.organ.com in 0.236494 Sec
         Set logs for peer1.opo.organ.com in 0.143125 Sec
         Set logs for peer0.hospital.organ.com in 0.207742 Sec
         Set logs for peer1.hospital.organ.com in 0.161781 Sec
Starting Step 2: Extract receive times.
         Done Extracting Receive Times in -17.989104 Sec
Stating Step 3: Extract validation times.
         Done Extracting Validation Times in -17.097077 Sec
Stating Step 4: Extract commit times.
         Done Extracting Validation Times in -8.190139 Sec
Starting Step 5: Return Propogation Time
         Processing peer0.histocompatibility.organ.com
         Processing peer1.histocompatibility.organ.com
         Processing peer1.opo.organ.com
         Processing peer0.gp.organ.com
         Processing peer0.hospital.organ.com
         Processing peer0.opo.organ.com
         Processing peer1.hospital.organ.com
         Processing peer1.gp.organ.com
Done
The length of propogation time is 146
```

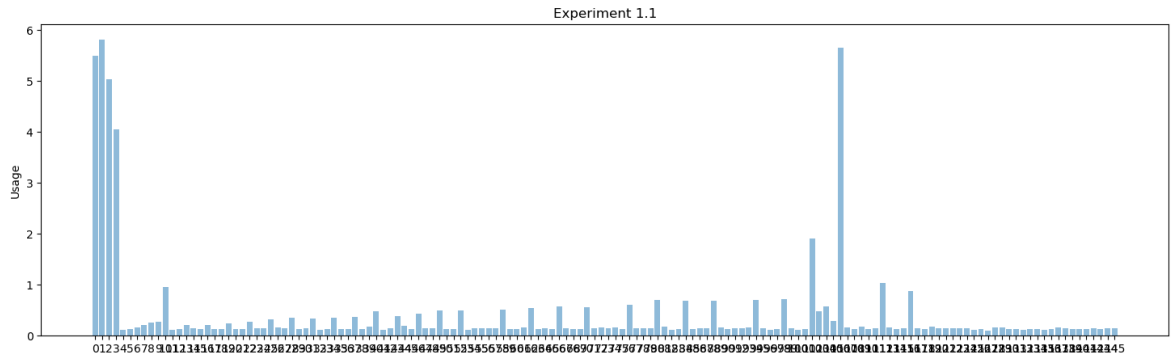
In [25]:
```
#Graphs
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

blocks = []
print("There are %s Blocks"%len(experiment4Results[0]))
for i in range(len(experiment4Results[0])):
    blocks.append(str(i))

propogation_time= experiment4Results[0]
y_pos = np.arange(len(propogation_time))
# performance = [10,8,6,4,2,1]
plt.subplots(figsize=(18,5))
plt.bar(y_pos, propogation_time, width=.8, align='center', alpha=0.5
)
plt.xticks(y_pos, blocks)
plt.ylabel('Usage')
plt.title('Experiment 1')

plt.show()
```

```
There are 146 Blocks
```

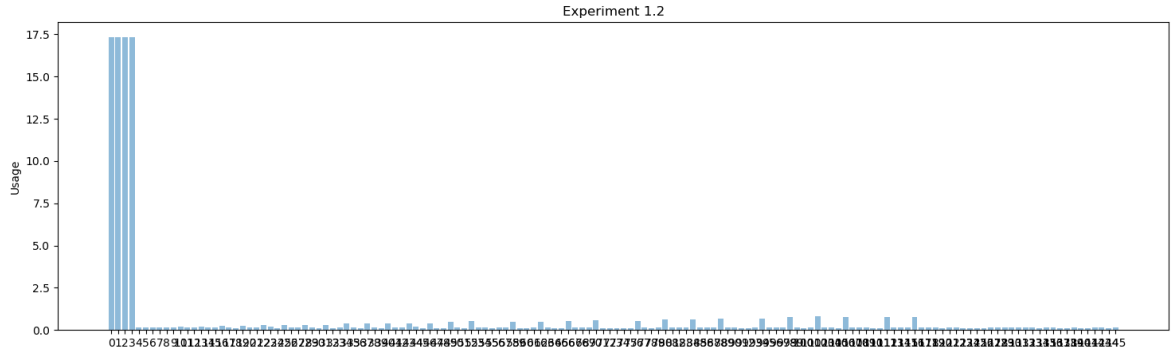


In [26]:
```
for j in range(1, 5):
    blocks = []
    print("There are %s Blocks"%len(experiment1Results[j-1]))
    for i in range(len(experiment1Results[j-1])):
        blocks.append(str(i))

    propogation_time= experiment1Results[j-1]
    y_pos = np.arange(len(propogation_time))
    # performance = [10,8,6,4,2,1]
    plt.subplots(figsize=(18,5))
    plt.bar(y_pos, propogation_time, width=.8, align='center', alpha=
0.5 )
    plt.xticks(y_pos, blocks)
    plt.ylabel('Usage')
    plt.title('Experiment 1.%s'% j)

    plt.show()
```

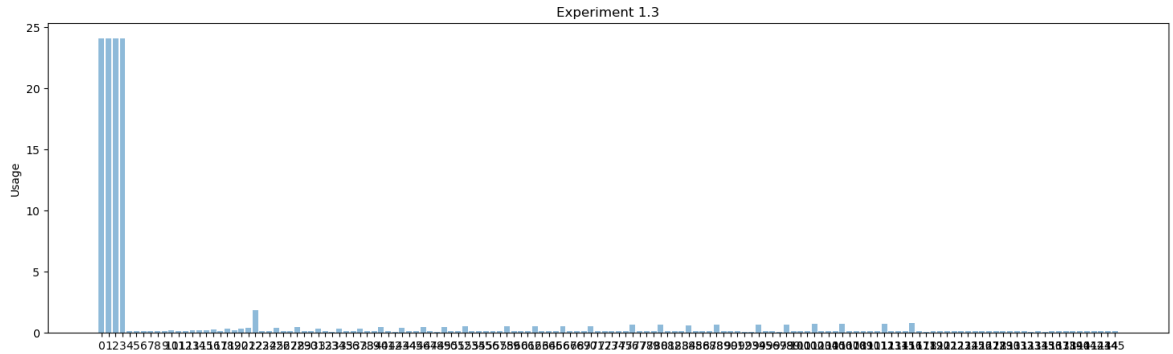
```
There are 146 Blocks
```




```
There are 146 Blocks
```



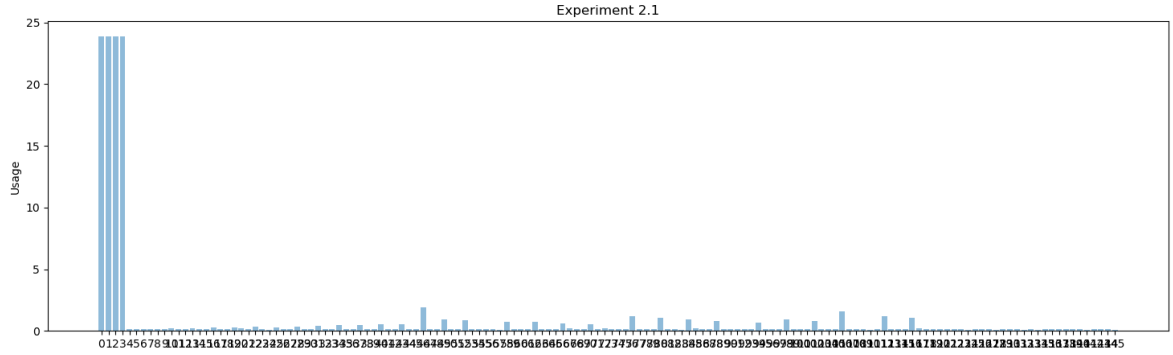

```
There are 146 Blocks
```



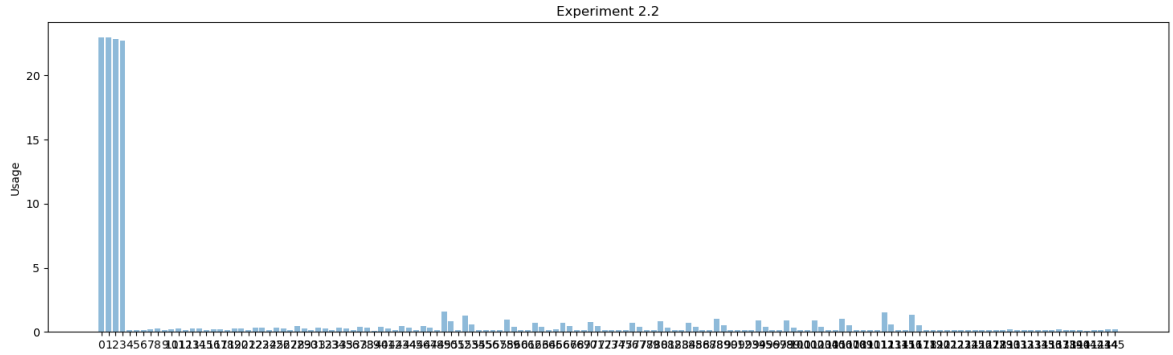

```
---------------------------------------------------------------------
------
IndexError                                Traceback (most recent call
last)
<ipython-input-26-334ef1656ad0> in <module>()
      1 for j in range(1, 5):
      2     blocks = []
----> 3     print("There are %s Blocks"%len(experiment1Results[j-1]))
      4     for i in range(len(experiment1Results[j-1])):
      5         blocks.append(str(i))

IndexError: list index out of range
```

In [35]:
```
for j in range(1, 5):
    blocks = []
    print("There are %s Blocks"%len(experiment2Results[j-1]))
    for i in range(len(experiment2Results[j-1])):
        blocks.append(str(i))

    propogation_time= experiment2Results[j-1]
    y_pos = np.arange(len(propogation_time))
    # performance = [10,8,6,4,2,1]
    plt.subplots(figsize=(18,5))
    plt.bar(y_pos, propogation_time, width=.8, align='center', alpha=
0.5 )
    plt.xticks(y_pos, blocks)
    plt.ylabel('Usage')
    plt.title('Experiment 2.%s'% j)

    plt.show()
```

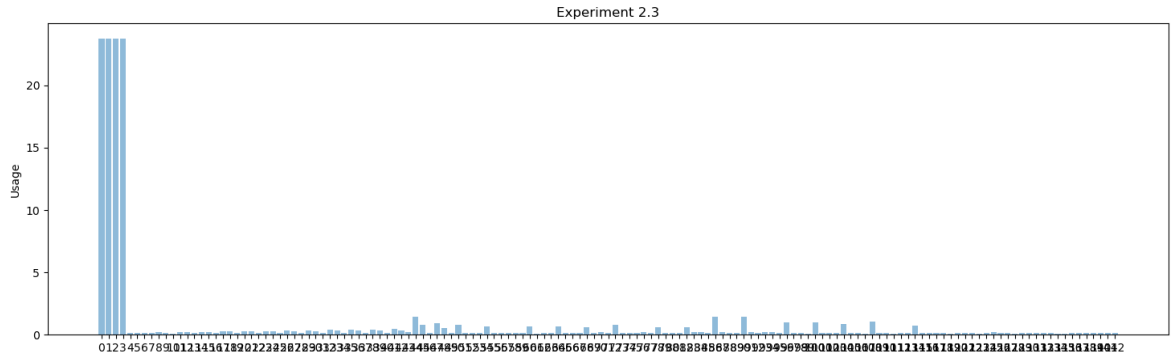
There are 146 Blocks



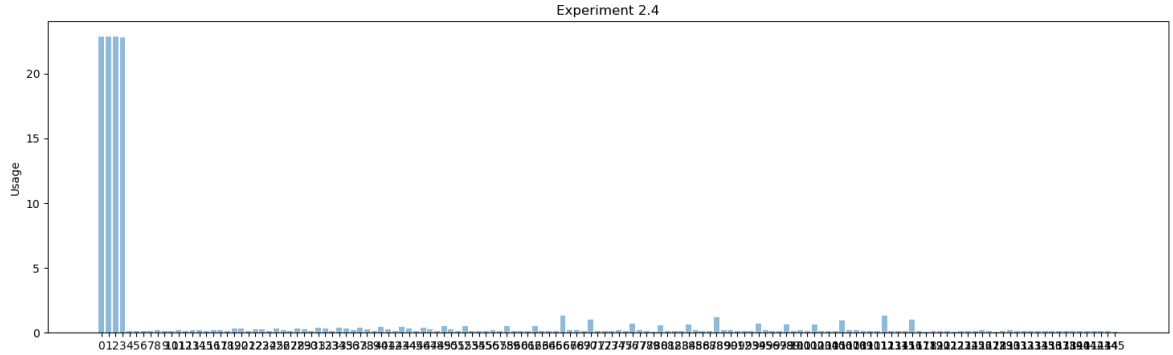
There are 146 Blocks



There are 143 Blocks

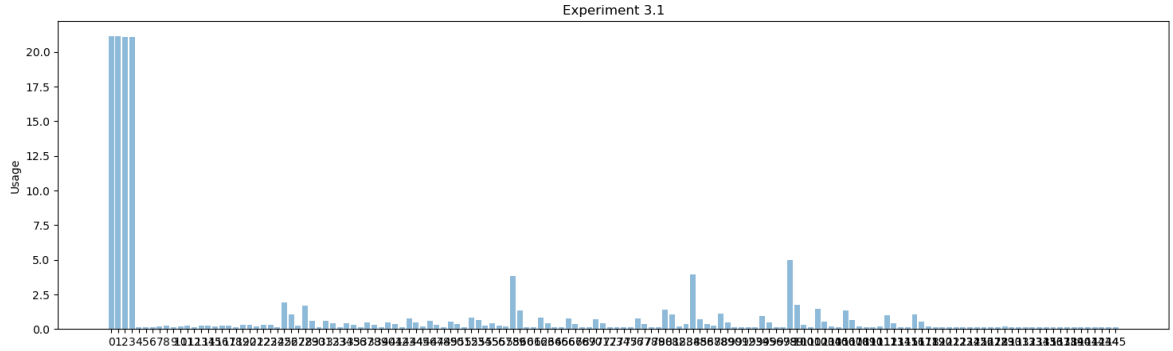


There are 146 Blocks

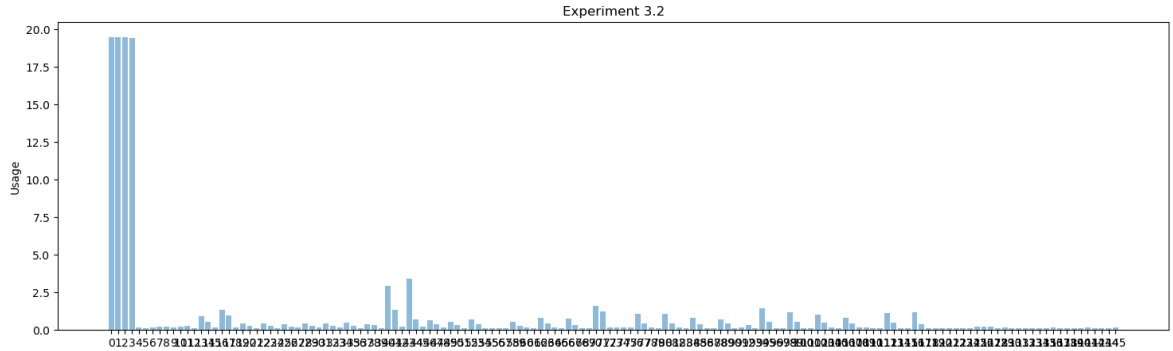


In [28]:
```
for j in range(1, 5):
    blocks = []
    print("There are %s Blocks"%len(experiment3Results[j-1]))
    for i in range(len(experiment3Results[j-1])):
        blocks.append(str(i))

    propogation_time= experiment3Results[j-1]
    y_pos = np.arange(len(propogation_time))
    # performance = [10,8,6,4,2,1]
    plt.subplots(figsize=(18,5))
    plt.bar(y_pos, propogation_time, width=.8, align='center', alpha=
0.5 )
    plt.xticks(y_pos, blocks)
    plt.ylabel('Usage')
    plt.title('Experiment 3.%s'% j)

    plt.show()
```

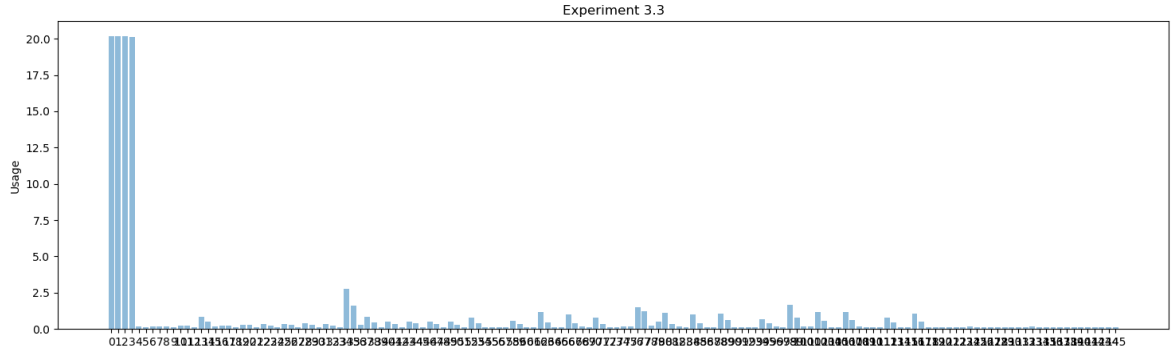
There are 146 Blocks



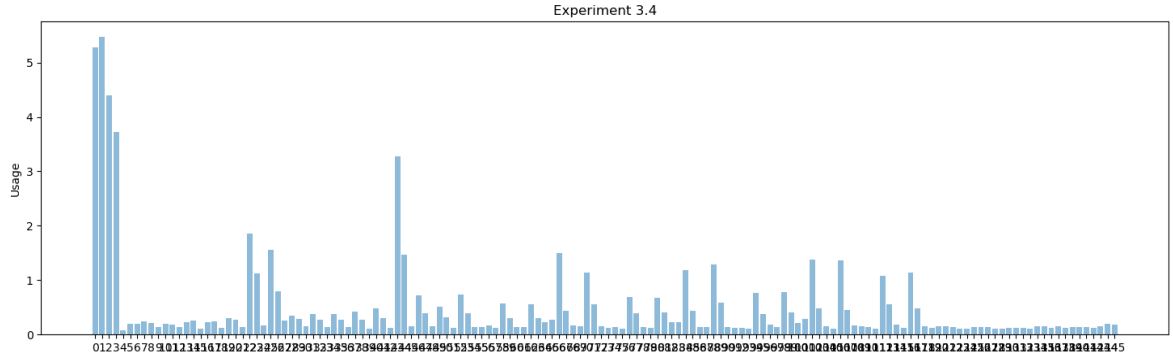
There are 146 Blocks



There are 146 Blocks

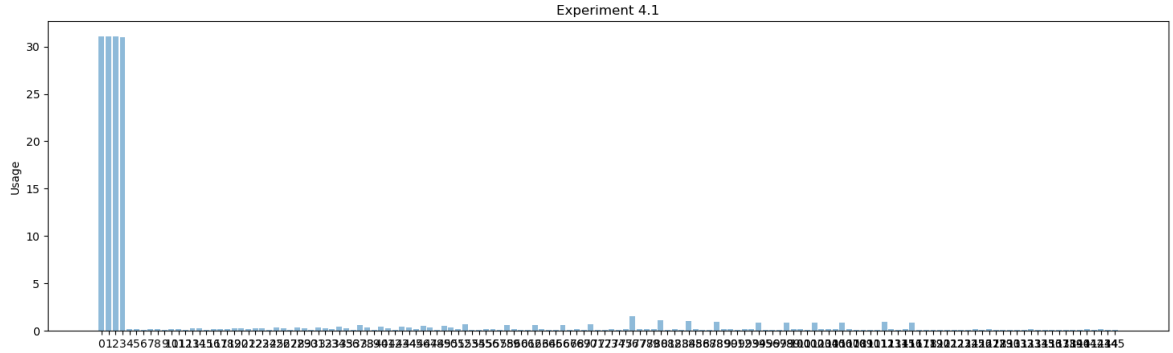


There are 146 Blocks

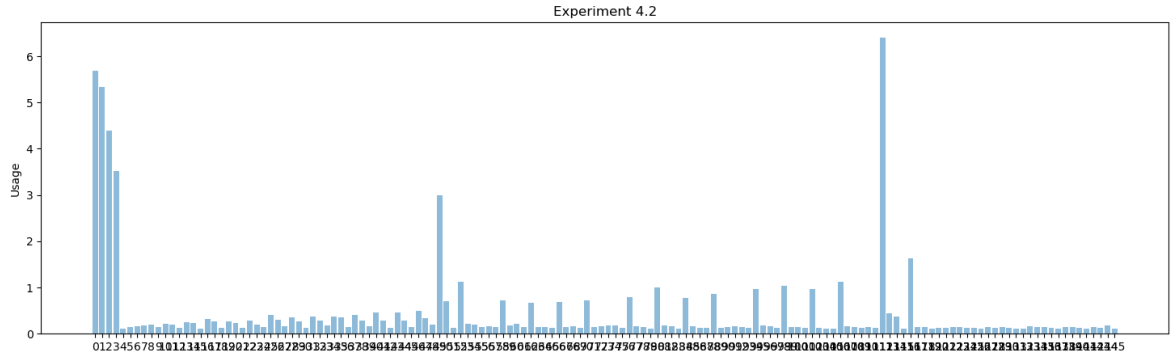


In [29]:
```
for j in range(1, 5):
    blocks = []
    print("There are %s Blocks"%len(experiment4Results[j-1]))
    for i in range(len(experiment4Results[j-1])):
        blocks.append(str(i))

    propogation_time= experiment4Results[j-1]
    y_pos = np.arange(len(propogation_time))
    # performance = [10,8,6,4,2,1]
    plt.subplots(figsize=(18,5))
    plt.bar(y_pos, propogation_time, width=.8, align='center', alpha=
0.5 )
    plt.xticks(y_pos, blocks)
    plt.ylabel('Usage')
    plt.title('Experiment 4.%s'% j)

    plt.show()
```

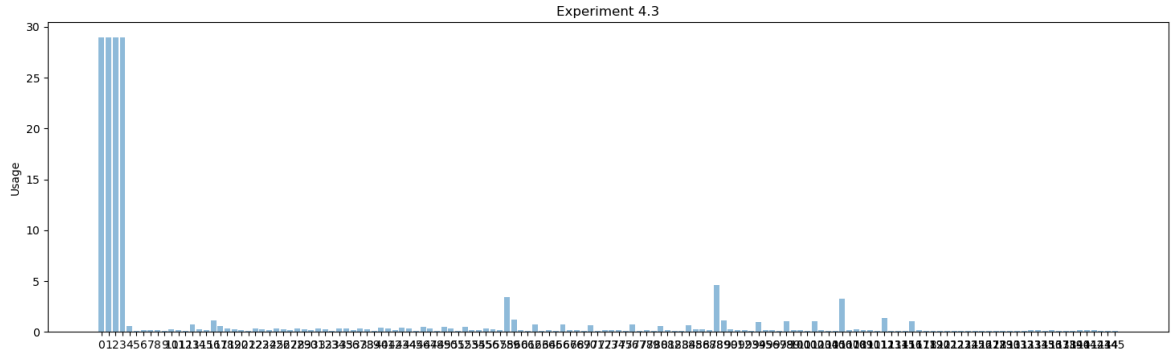
There are 146 Blocks



There are 146 Blocks



There are 146 Blocks



There are 146 Blocks