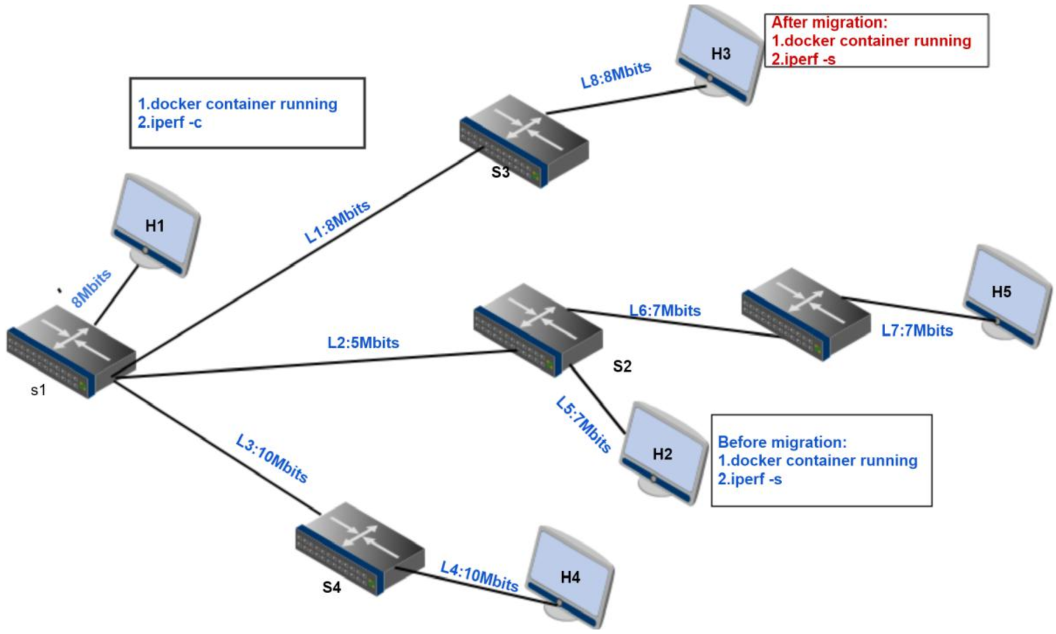
Policy Based Docker migration Experiment

Experiment of different policies

Goal: Policy based Docker Container migration in SDN environment, by collecting results and doing the plot, do the comparison and analyse different policies.

Graph1: Bandwidth and Shortest Path make the same decision (verify Bandwidth performance is compatible with shortest path)



Why choose this topology:
- ✧ L2 can be a bottleneck link for the bandwidth
- ✧ L2 can be a factor of choosing the shortest path
- ✧ Compare the performance of different policy when bandwidth and shortest path make the same decision

Note:
All the migration destination is chosen in the free host set(no docker container running)

- ## Use Cases

Use case1:
Policy: bandwidth
Scenario: H1 H2 run Docker container which hosts iperf server service
Traffic generation: H1 as iperf client, H2 as iperf server
Migration Analysis: using **Max(min(bandwidth))** select one has the most available bandwidth

H2 is detected as heavy host(migration source host)
H2 -> H3 , path bandwidth=5
H2 -> H4 , path bandwidth=5

H2 -> H5, path bandwidth=7
So H5 is selected as migration destination host
Migration decision: H2->H5

Use case2:
Policy: shortest path
Scenario:H1 H2 run Docker container which hosts iperf server service
Traffic generation: H1 as iperf client, H2 as iperf server
Migration analysis: using floodlight REST API query the fastest path,according to the path latency
H2 is detected as heavy host(migration source host)
H2 -> H3 , path H2->S2->S1->S3->H3
H2 -> H4 , path h2->S2->S1->S4->H4
H2 -> H5 , path h2->S2->S5->H5
So H5 will be considered as the shortest path with least latency
Migration decision: H2 -> H5

Use Case3:
Policy: random
Scenario: H1 H2 run Docker container which hosts iperf server service.
Traffic generation: H1 as iperf client, H2 as iperf server.
Migration Analysis: randomly select migration source host from busy host set,
migration destination host from free host set.
busy host set (H1, H2)
Free host set (H3,H4,H5)
Migration decision: H1 -> H3 , H1 -> H4 , H1 -> H5
H2 -> H3 , H2 -> H4 , H2 -> H5

- ## Plots set

Where does these plots come from?
->For bandwidth consumption plot,the value is the sum of send and receive bandwidth consumption ,time starts before migration occur. It experiences migration process,and some time after migration done. The plot clearly showed the bandwidth consumption changes before, during,and after migration.
->For thoughput accumulation plot, the value is the number of bytes transfer along with the time. It showed that the totally traffic during the migration process.

Figure1(Use Case1)

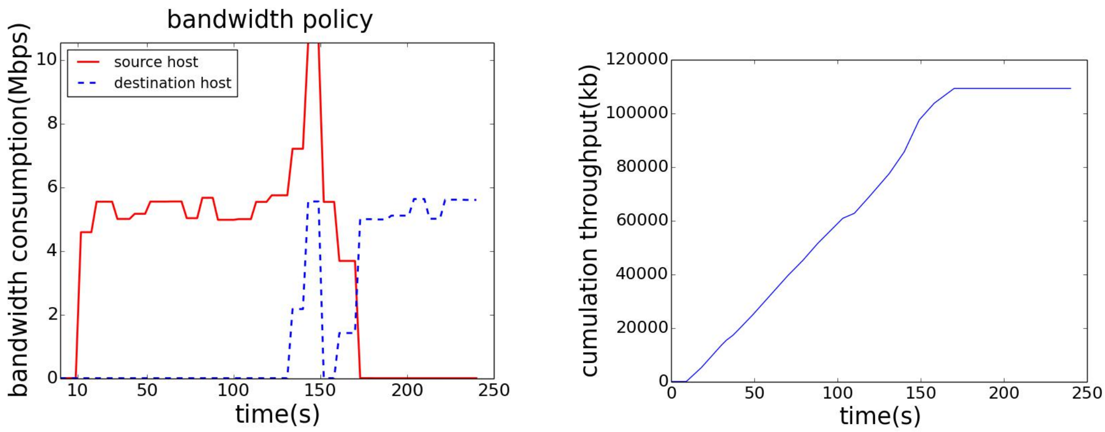


Figure2(Use Case2):

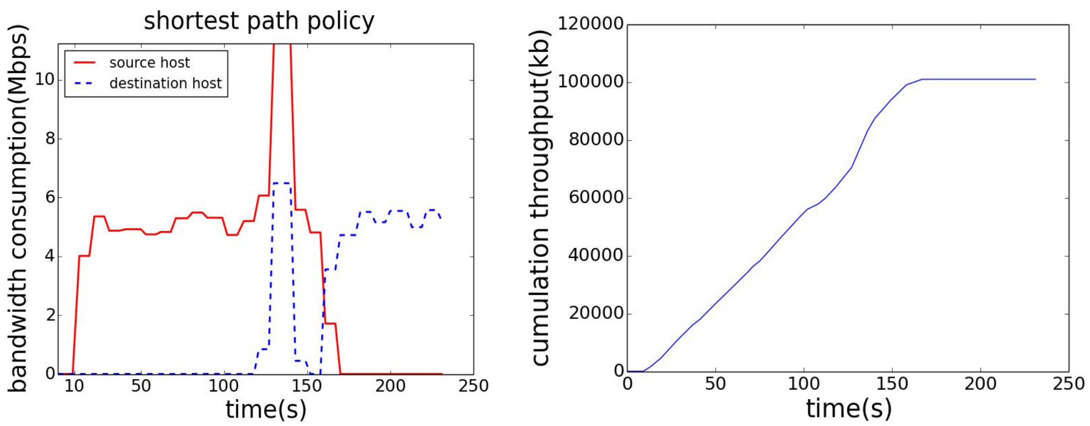
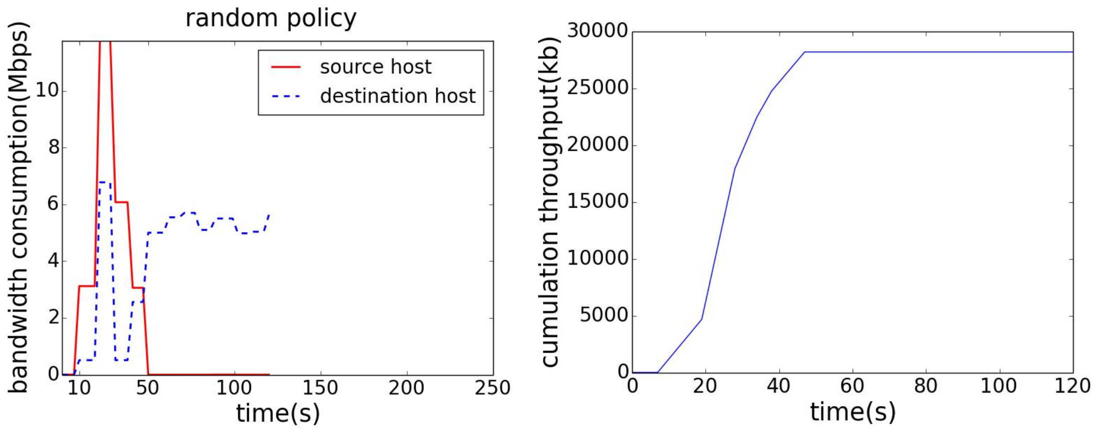


Figure3(Use Case3):



From those figures, we can see that all the migration policy have the same change shape.The difference is the traffic quantity and the time when it starts the migration. Analyzing those figures, we can characterize 3 period:

First period, the traffic data is collected before migration process from the system start running, so for the red line(source host) initially it just running the docker container which acts as iperf server, it receive and send data to iperf client.

second period, as the time going after first period, the iperf traffic generating is increasing. Then the switch source host attached is detected as heavy switch and the host is selected to be the migration source host. Then, It starts the migration process. During this period, the source host not only generates traffic with the iperf client, but also has the traffic with the migration destination host for docker image file transfer. As for the blue line(destination host), it start to receive the migration file, so bandwidth consumption is starting increase during this interval. Then after migration is done, it enters the third period.

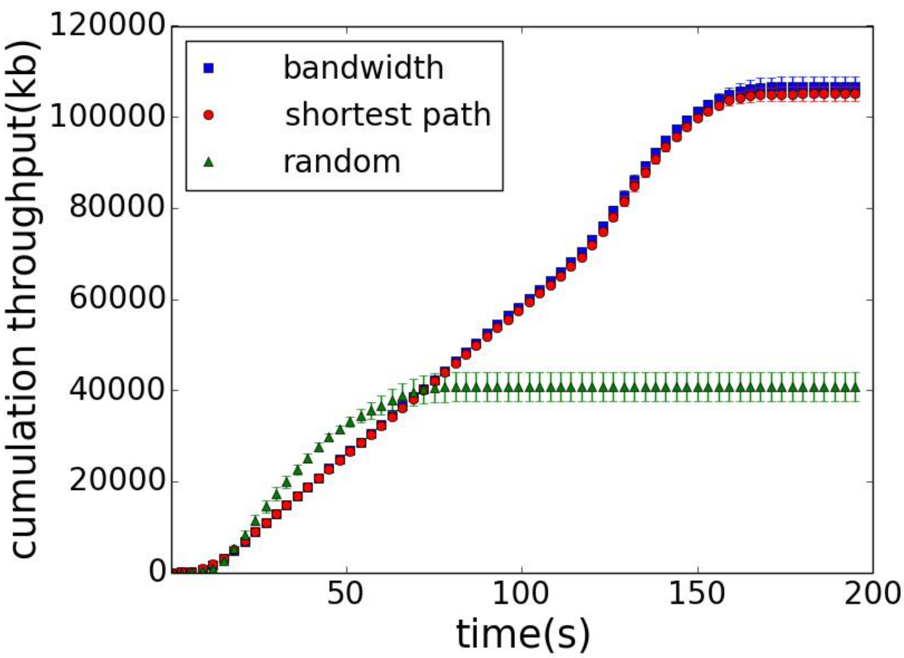
Third period, the red line(source host), no more running the iperf server,so there is no traffic anyone(since here we only have iperf server,no other server). Instead of blue line(destination host), it starts to run the iperf server after migration, it will have traffic from the iperf client,So the bandwidth consumption for source host is increasing continuously.

- **Thoughput Accumulation Comparison**

Where does these data come from?

The traffic is collected from migration source host before migration process starting. It stops when the migration is done, here the host only runs iperf server, so when the server is migrated from the host, there is more traffic anymore, the cumulation throughput enter a stable stage.

Figure4(cumulation throughput combination)



Vertical comparison:

It shows that in the first 50s, the throughput accumulation of random policy is increasing faster than bandwidth and shortest path policy. It implies that system starts migration process earlier when using random policy than bandwidth and shortest path policy.when random migration occurs, the network situation is not too bad, and the migration source host doesn't suffer a high load of traffic. After 50s, shortest path already finished the migration process, the accumulation doesn't increase anymore. Instead of bandwidth and shortest path, it continue increasing, this means the network traffic is heavier and heavier until when it is detected, and migration is done. Bandwidth and shortest path has larger number, it means that when migration occurs, the migration source host is suffering heavy traffic load. There is much more traffic in the network.

Horizon comparison:

Here we compare the saturation point, which means the migration process complete time. It can be seen clearly that random policy arrives much earlier than bandwidth and shortest path policy. When the network has high requirement of latency, for example, migration must be done in 50s, it is better to choose random policy.

As we can see, when the migration decision made by shortest and bandwidth policy is the same, their performance are compatible with each other.

Accumulation confidence Interval

The migration decision made by bandwidth and shortest path are determinate. for bandwidth H2->H3, for shortest path H2->H3. So each run in the same network condition the thoughput accumulation value varies lightly. As can be seen in figure4, the error bar is very small is very small. Instead of Random, the migration destination is not determinate, so each run, it may choose different migration decision, the accumulation value at each time point is different. Hence, the error bar is bigger comparing with bandwidth and shortest path policy.

- **Migration Time Comparison**

Table1(migration time 10.0.0.2-10.0.0.5)

	bandwidth	shortest path	random
1	12s	12s	30s
2	12s	37s	29s
3	12s	13s	12s
4	14s	17s	33s
5	15s	17s	34s
6	15s	28s	12s
7	15s	17s	33s

8	17s	28s	37s
9	37s	19s	12s
10	18s	21s	35s
11	33s	19s	35s
12	18s	20s	12s
13	31s	19s	31s
14	18s	20s	31s
15	29s	19s	12s
16	19s	20s	38s
17	27s	20s	38s
18	24s	20s	12s
19	25s	20s	35s
20	25s	20s	12s
average	(20.8 ±2.7)s	(20.3 ±1.9) s	(26.15 ±3.91) s

Table2(migration time random policy based)

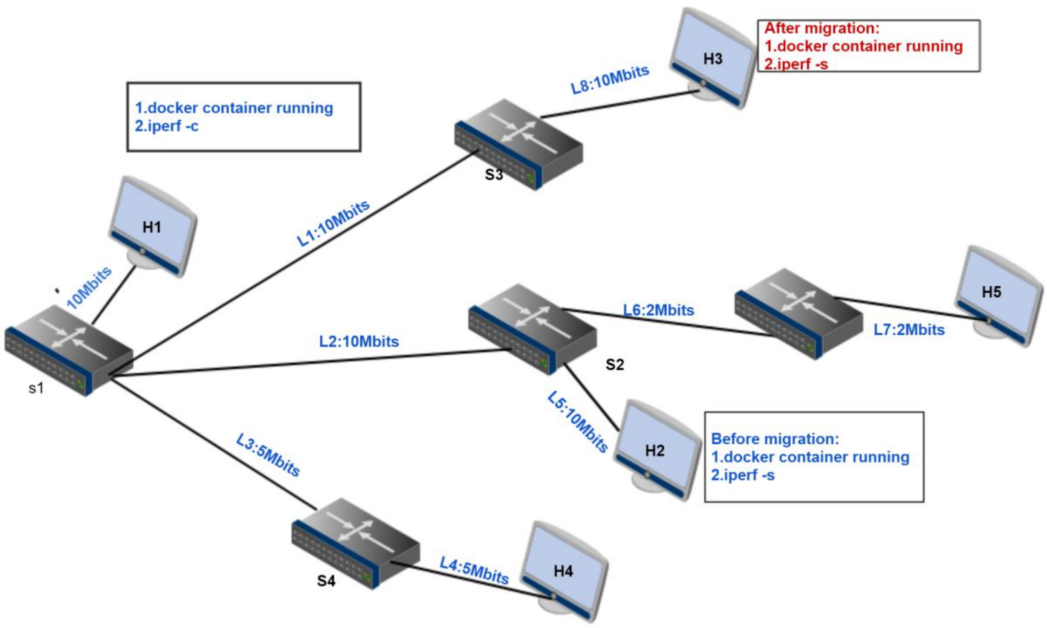
	10.0.0.2-10.0.0.3	10.0.0.2-10.0.0.4	10.0.0.2-10.0.0.5
1	30s	29s	12s
2	33s	34s	12s
3	37s	37s	12s
4	31s	29s	12s
5	35s	35s	12s
6	38s	38s	12s
7	31s	32s	12s
8	35s	35s	12s
9	38s	38s	12s
10	31s	32s	12s
11	35s	36s	12s
12	43s	39s	12s
13	32s	33s	12s
14	35s	36s	12s
15	32s	34s	13s
16	36s	36s	13s
17	32s	34s	13s
18	37s	36s	13s
19	33s	34s	13s
20	37s	37s	13s
average	(34.6 ±1.2) s	(34.7 +1.0) s	(12.3+0.2) s

From table1, it can be seen that the migration time between bandwidth, shortest path, random,from the same source host 10.0.0.2 to the same destination host 10.0.0.5, concerning the average migration time, random has the smallest migration time. Shortest path,and bandwidth they are almost the same.
Behind story: For bandwidth and shortest path policy, the main goal is load balancing,the system first detect the heavy switch, then apply the policy algorithm, so in this case, the available bandwidth for migration will be much smaller, this results in larger migration time. Instead of random, it doesn't concern the load of each switch, just simply choose the source and destination. Most of the case, before the switch getting heavy, it starts the migration process,so
the available bandwidth is larger, this results in smaller migration time.

From table2, it can be seen that the migration time various under policy random while migrating to different host. It proves that L2 is the bottleneck link with small bandwidth. When migration to 10.0.03 or 10.0.0.4, it always needs larger migration time. It implies that random policy has uncertainty, this feature can be used to do simple attack defence. But in terms of the migration performance, it can not guarantee the best migration destination.

Combine table1 and table2, we can conclude that shortest path and bandwidth can do load balancing, after migration, it can guarantee the better service performance. But under switch overloading, migration process takes longer time. Random policy is good for attack defence, but it concerns less the traffic situation. It does migration even when switch is not overloaded.

Graph2: Bandwidth and Shortest Path make the different decision (Verify Bandwidth performance is better than Shortest path)



Why choose this topology:

- ✧ Compare the performance of different policies when bandwidth and shortest path make the different decision
- ✧ Verify the performance of bandwidth is better than shortest path
- ✧ Network link band width varies a lot

- **Use Cases**

Use case1:
Policy: bandwidth
Scenario: H1 H2 run Docker container which hosts iperf server service
Traffic generation: H1 as iperf client, H2 as iperf server
Migration Analysis: using **Max(min(bandwidth))** select one has the most available bandwidth

H2 is detected as heavy host(migration source host)
H2 -> H3 , path bandwidth=10
H2 -> H4 , path bandwidth=5
H2 -> H5, path bandwidth=2
So H3 is selected as migration destination host

Migration decision: H2->H3

Use Case2
Policy: shortest path
Same as use case2 in Graph1
Migration Decision: H2->H5

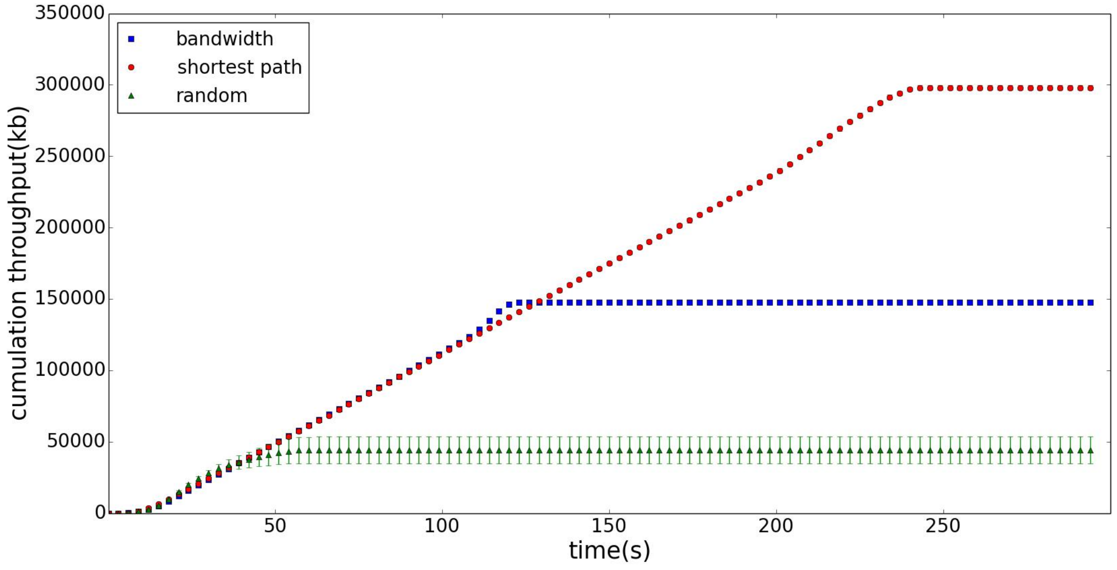
Use Case3
Policy: random
Same as use case3 in Graph1
Migration decision: H1 -> H3 , H1 -> H4 , H1 -> H5
H2 -> H3 , H2 -> H4 , H2 -> H5

- **Plots set**

The migration bandwidth consumption plot for each policy is same with in Graph1, except the value varies according to the time.

- **Thoughput Accumulation Comparison**

Figure5: cumulation throughput combination



Vertical comparison:

the same conclusion as in Graph1

Horizon comparison:

Here we compare the saturation point, which means the migration process complete time. It can be seen clearly that random policy arrives much earlier than bandwidth and shortest path policy at about 50s. Then bandwidth finish the migration process at about 120s, which is earlier than shortest path.

Random < bandwidth < shortest path

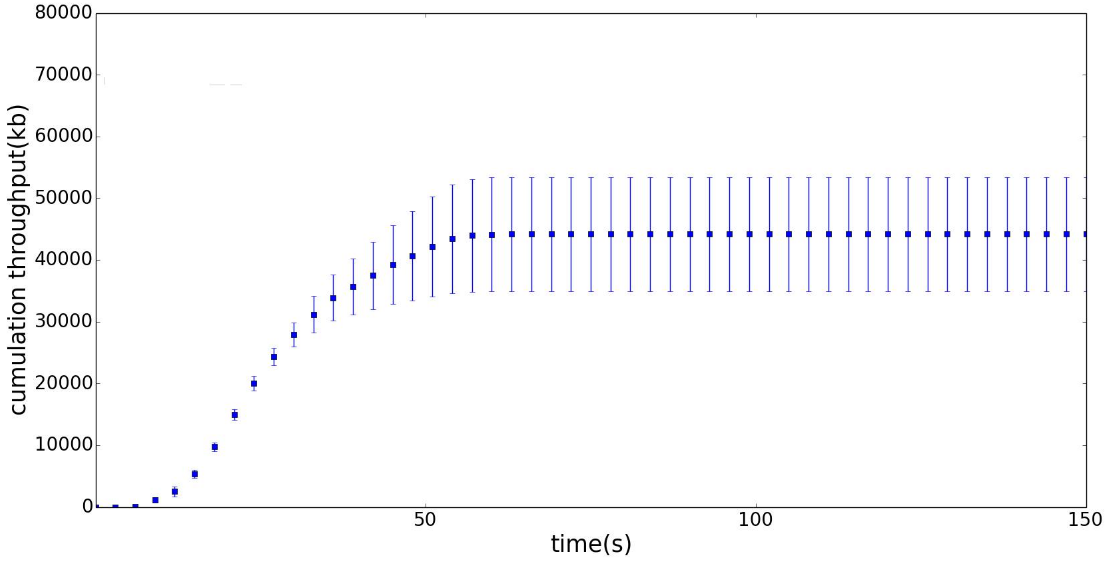
When the network has high requirement of latency, for example, migration must be done in 50s, it is better to choose random policy. If we want to chose one policy considering the trade off of migration finish time and load balancing. It is better to choose bandwidth policy.

As we can see, when the link bandwidth is much bigger than the shortest path link, the bandwidth performance is better than shortest path.

Accumulation confidence Interval

The migration decision made by bandwidth and shortest path are determinate. for bandwidth H2->H3, for shortest path H2->H5. So each run the value is very similar. As can be seen in figure5, the error bar is very small nearly invisible. Instead of Random, the migration destination is not determinate, so each run, it may choose different migration decision, the accumulation value at each time point is different. Hence, the error bar is bigger. Figure6 shows clearly.

Figure6:Random Policy Accumulation



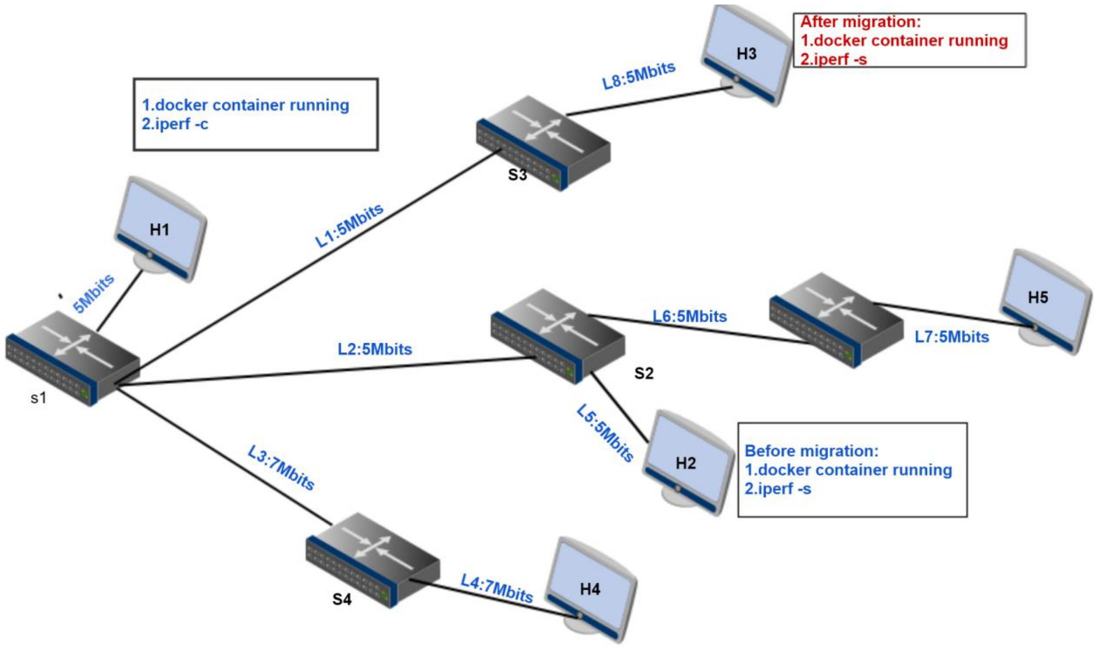
- **Migration Time Comparison**

	bandwidth	shortest path	random
1	9s	40s	9s
2	9s	40s	9s
3	9s	40s	9s
4	9s	40s	16s
5	9s	40s	16s
6	9s	40s	17s
7	9s	40s	40s
8	9s	40s	40s
9	9s	40s	40s
10	10s	40s	40s
average	(9.1 ±0.15)s	(40.0 ±0) s	(23.6 ±7.12) s

This table implies that if the link bandwidth change, it effect the migration time greatly for different policies. Here bandwidth policy take the advantage of the bigger link bandwidth, so the migration time is much smaller than shortest path, and random. For the confidence interval, random policy varies a lot. Because different migration destination path experience different path available bandwidth.

Combine all the features of Granph2, if the network topology link bandwidth difference is significant big. From different requirement aspects, if user wants the sooner migration finish time, random policy is considered. If user wants do network load balancing and minimize the migration time, bandwidth policy is considered.

Graph3: Bandwidth and Shortest Path make the different decision
And bandwidth decision non determinate
(Verify Shortest Path performance is better than bandwidth path)



Why choose this topology:
- ✧ Compare the performance of different policies when bandwidth and shortest path make the different decision,and bandwidth decision not determinate
- ✧ Verify the performance of shortest path is better than bandwidth
- ✧ Network link bandwidth evenly

- **Use Cases**

Use case1:
Policy: bandwidth
Scenario: H1 H2 run Docker container which hosts iperf server service
Traffic generation: H1 as iperf client, H2 as iperf server
Migration Analysis: using **Max(min(bandwidth))** select one has the most available bandwidth

H2 is detected as heavy host(migration source host)
H2 -> H3 , path bandwidth=5
H2 -> H4 , path bandwidth=5
H2 -> H5, path bandwidth=5
So H3, H4, or H5 may be selected as migration destination host

Migration decision: H2->H3, or H2->H4, or H2,H5

Use case2:
Policy: shortest path
Same as in graph1 and graph2
Migration decision: H2 -> H5

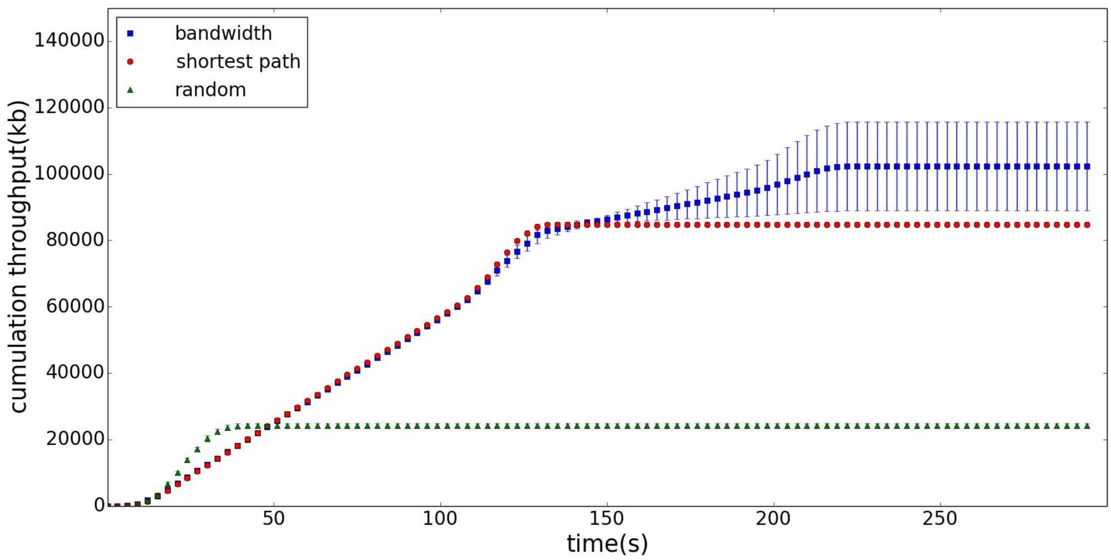
Use Case3:
Policy: random
Scenario: H1 H2 run Docker container which hosts iperf server service.
Traffic generation: H1 as iperf client, H2 as iperf server.
Migration Analysis: randomly select migration source host from busy host set,
migration destination host from free host set.
busy host set (H1, H2)
Free host set (H3,H4,H5)
Migration decision: H1 -> H3 , H1 -> H4 , H1 -> H5
H2 -> H3 , H2 -> H4 , H2 -> H5

- **Plots set**

The migration bandwidth consumption plot for each policy is same with in Graph1, except the value varies according to the time.

- **Thoughput Accumulation Comparison**

Figure7: cumulation throughput combination



Vertical comparison:

the same conclusion as in Graph1

Horizon comparison:

Here we compare the saturation point,which means the migration finishing time.

Random < Shortest Path < bandwidth

When the network has high requirement of latency, for example, migration must be done in 50s, it is better to choose random policy. If we want to chose one policy considering the trade off of migration finish time and network load balancing. It is better to choose bandwidth policy.

As we can see, when the link bandwidth is evenly distributed, probably there are multiple migration destination when using bandwidth policy. If shortest path policy choose the destination has the equivalent bandwidth with bandwidth, shortest path is better.

Accumulation confidence Interval

The migration decision made by bandwidth is not determinate. For bandwidth,the decision could be H2->H3, H2->H4, H2->H5. So when the migration decision will be different, the network traffic condition will also be different, in the figure, we can see that the error bar for bandwidth is bigger than shortest path and random. Instead, for random policy, here we can see that confidence interval is very small.

- **Migration Time Comparison**

	bandwidth	shortest path	random
1	17s	17s	17s
2	17s	17s	17s
3	17s	17s	17s
4	20s	17s	17s
5	21s	17s	17s
6	18s	17s	17s
7	18s	17s	17s
8	19s	17s	17s
9	17s	17s	17s
10	22s	18s	18s
average	(18.6 ±0.91)s	(17.1 ±0.16) s	(17.1±0.16) s

This table implies that when the link bandwidth is homogeneous in the network with a smaller value. It cause bandwidth policy has multiple choices. So the migration destination is not determinate, and it cause the migration time is also different when concerning the load balancing. And in this case, shortest path, and random, they use compatible migration time.

Combine all the features of Granph2, if the network topology link bandwidth is homogeneous, shortest path perform better than bandwidth. From different requirement aspects, if user wants the sooner migration finish time, random policy is considered. If user wants do network load balancing and minimize the migration time, shortest policy is considered.

- ## Conclusion

Combine Graph1, Graph2,Graph3, we can get that it is hard to define which policy is the best one. As in the experiment, the topology is the same for graph1, graph2, graph3, the link bandwidth is different. This 3 graph represent 3 types network. For each graph, different policies has different performance. So when the user use the migration system. It is better to analysis the topology a little bit, then choose a suitable policy. On the other hands, it is also depends on the user prevision, requirements, and needs of the network. If it is used to do simple defence, and have high requirement to the migration finishing time, it is better to use random policy. If it concerns network load balancing and migration time, it is better choose shortest path, or bandwidth.