mathematics

MDPI

Article

Constructing Two Edge-disjoint Hamiltonian Cycles on BCube Data Center Networks with Applications to All-to-all Broadcasting

Kung-Jui Pai[1,*]

[1] Department of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City 24301, Taiwan; poter@mail.mcut.edu.tw
* Correspondence: poter@mail.mcut.edu.tw

Appendix 1. Supplementary Materials

The Python implementations of the two algorithms proposed in this paper.

```
def printv(xx, n):
    for i in range(len(xx)):
        print(xx[i], end='-')
        if i % n == n-1: print()

# first EDHC in BCube(n,1) while n is even
n = 4

v = [i for i in range(n-1,-1,-1)] # n-1, n-2, ... , 2, 1, 0
print(v)
list1=[]
for g in range(n):
    for x in v:
        list1.append(int(str(g)+str(x)))
    v.reverse()
printv(list1, n)

# second EDHC in BCube(n,1) while n is even
v1 = [i for i in range(1, n, 2)] # 1-3-5-…-(n – 3)-(n – 1)
v2 = [i for i in range(0, n, 2)] # 0-2-4-…-(n – 4)-(n – 2)
v = v1 + v2
print(v)

list2=[]
for g in range(n):
    for x in v:
        list2.append(int(str(g)+str(x)))
    v.reverse()
printv(list2, n)
```

```
def printv(xx, n):
    for i in range(len(xx)):
        print(xx[i], end='-')
        if i % n == n-1: print()

# first EDHC in BCube(n,1) while n is odd
n=5
v = [i for i in range(n-1,-1,-1)] # n-1, n-2, ... , 2, 1, 0
print(v)
list1=[0]
for x in range(n-1,0,-1): # visit all vertices in the first BCube(n,0)
    list1.append(x)
for x in range(1,n): # visit all vertices except vertex 10 in the second BCube(n, 0)
    list1.append(int(str(1)+str(x)))
#printv(list1, n); print(); print()
for g in range(2,n):
    for x in v:
        list1.append(int(str(g)+str(x)))
    v.reverse()
list1.append(10) # visit vertex 10
printv(list1, n)

# second EDHC in BCube(n,1) while n is odd
cntr =int(n/2); list2=[cntr] # start from cntr
v = [(cntr-1-i*2)%n for i in range(n)] # (n/2 – 1)-(n/2 – 3)-(n/2 – 5)-...-(n/2+3)-(n/2+1)
print('center=',cntr, ' , v=',v)

for x in range(1,n):          # walk along (x, x-2), visit all vertices in 1st BCube
    list2.append((cntr - 2*x) % n)
for i in range(1,int(n/2)+1): # visit (n/2) vertices in 2rd BCube(n, 0)
    x = (cntr + 2*i ) % n
    list2.append(int(str(1)+str(x)))

for g in range(2,n): #
    for x in v:
        list2.append(int(str(g)+str(x)))
    v.reverse()

for i in range(int(n/2)+1, n+1): # visit remaining (n/2)+1 vertices in 2rd BCube(n, 0)
    x = (cntr + 2*i ) % n
    list2.append(int(str(1)+str(x)))

printv(list2, n)
```

We conducted broadcast simulations on $BCube(n, 2)$ with $3 \leq n \leq 9$. Tables 1 and 2 report the experimental results for *ABL* and *MBL*, corresponding to the approaches using a single Hamiltonian cycle and two EDHCs, respectively.

Table 1. *ABL* and *MBL* using one Hamiltonian cycle for broadcasting in *BCube*(n, 2) while $3 \leqslant n \leqslant 9$.

BCube(n, 2)	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$
ABL	1061.09	1081.45	1110.72	1156.31	1220.58	1304.74	1413.71
MBL	2123	2160	2221	2312	2439	2608	2825

Table 2. *ABL* and *MBL* using two EDHCs for broadcasting in *BCube*(n, 2) while $3 \leqslant n \leqslant 9$.

BCube(n, 2)	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$
ABL	540.41	563.91	603.28	658.62	740.05	843.99	981.77
MBL	1074	1111	1172	1263	1390	1559	1776

We combine the data from Tables 1 and 2 to plot Figure 9, which facilitates comparison of ABL and MBL under two approaches: a single Hamiltonian cycle and two EDHCs.

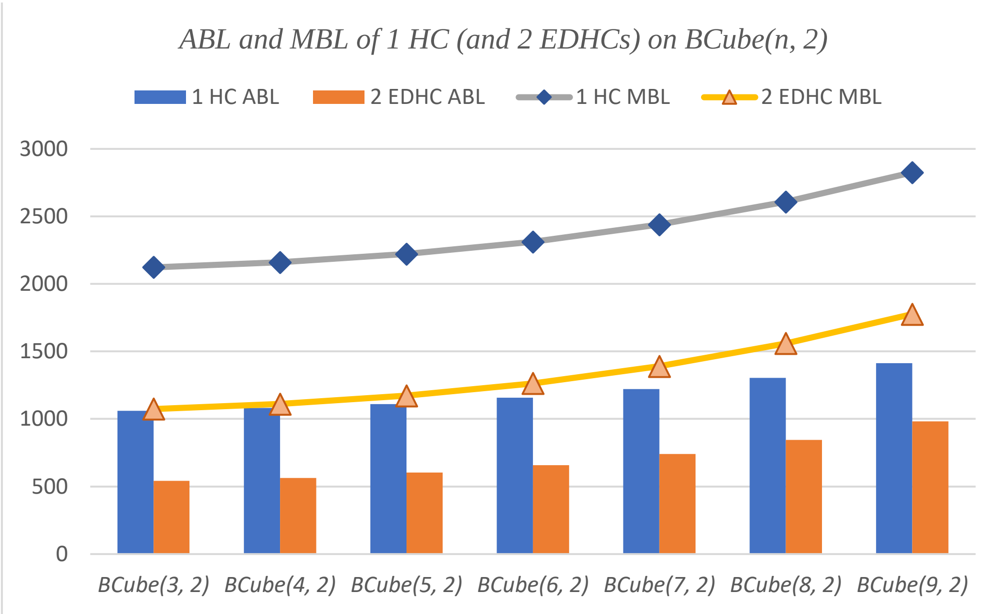

Figure 9. The comparisons of *ABL* and *MBL* between one Hamiltonian cycle and two EDHCs as the broadcasting channels on *BCube*(n, 2) while $3 \leqslant n \leqslant 9$.

We conducted broadcast simulations on *BCube*(n, 1) where $3 \leq n \leq 9$. However, the number of vertices in these graphs ranges only from 9 to 81, and the resulting *ABL* and *MBL* are very similar. Therefore, we present these results in here. Tables 3 and 4 report the experimental results for *ABL* and *MBL*, corresponding to the approaches using a single Hamiltonian cycle and two EDHCs, respectively.

Table 3. *ABL* and *MBL* using one Hamiltonian cycle for broadcasting in *BCube*(n, 1) while $3 \leqslant n \leqslant 9$.

BCube(n, 1)	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$
ABL	1053.22	1056.63	1060.87	1066.34	1073.08	1080.63	1088.86
MBL	2105	2112	2121	2132	2145	2160	2177

Table 4. *ABL* and *MBL* using two EDHCs for broadcasting in *BCube*(n, 1) while $3 \leqslant n \leqslant 9$.

BCube(n, 1)	$n = 3$	$n = 4$	$n = 5$	$n = 6$	$n = 7$	$n = 8$	$n = 9$
ABL	528.9	535.37	539.23	546.16	554.52	563.92	574.51
MBL	1056	1063	1072	1083	1096	1111	1128

We combine the data from Tables 3 and 4 to plot Figure 10, which facilitates comparison of ABL and MBL under two approaches: a single Hamiltonian cycle and two EDHCs.

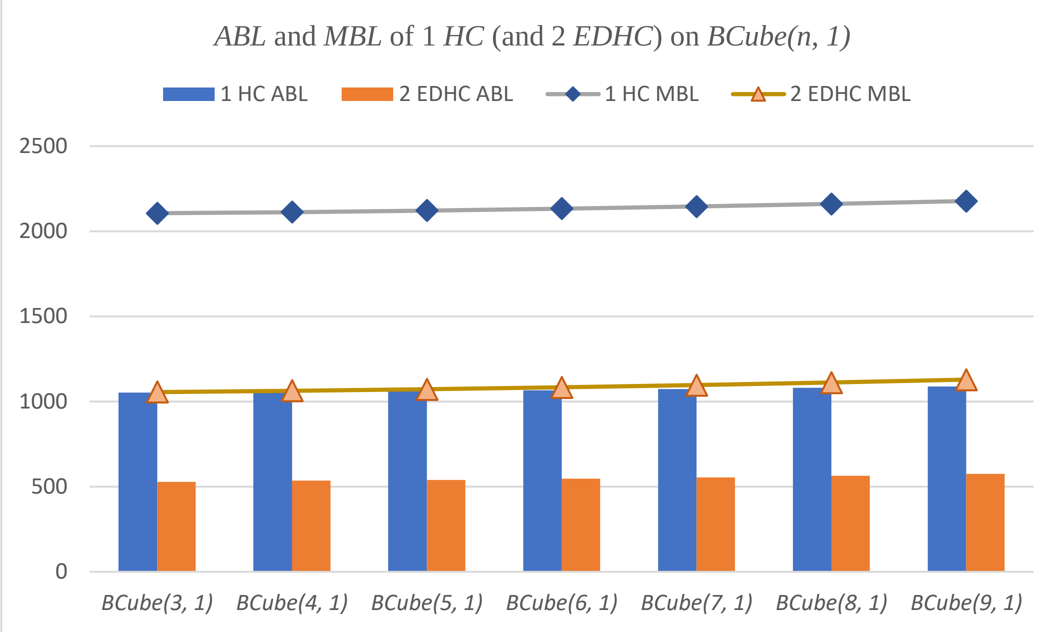

Figure 10. The comparisons of *ABL* and *MBL* between one Hamiltonian cycle and two EDHCs as the broadcasting channels on *BCube*(n, 1) while $3 \leqslant n \leqslant 9$.