



[22.10.14] 다운로드서비스

# T5 다운로드서비스

김 태 현

# 문 제

- 3가지 요소를 가지는 트리 존재
  - 1) 서버 : 트리의 root, 전송 속도 capa로 데이터 제공
  - 2) PC : 트리의 leaf, 원하는 크기만큼 데이터를 다운로드 하고 연결 해제
  - 3) 허브 : 서버 또는 부모 허브에서 받은 데이터를 자식 허브 또는 PC에 전달
- 서버 또는 허브는 자신과 연결된 자식 허브 또는 PC 중, 다운로드 필요한 자식 허브 또는 PC에 균등하게 전송 속도 분배 소수점 이하는 버림
- PC가 추가되거나 다운로드 완료하여 자동으로 접속 해제될 때, 즉시 다운로드 서비스 전송 속도 변경

## ※ 제약사항

- Time은 호출시마다 1 이상 증가 ( $1 \leq \text{Time} \leq 50,000$ )
- 서버 또는 허브가 가질 수 있는 자식 허브(pc포함)의 개수는 5개 이하
- addHub()  $\leq 10,000$  , removeHub()  $\leq 100$   
requestDL()  $\leq 1,000$ , checkPC()  $\leq 1,000$

## void init(int capa)

시각 0에 ID가 0인 서버가 capa의 전송 속도로 존재

## void addHub(int Time, int parID, int chID)

Time 시각에 parID인 서버 또는 허브에 chID 자식 허브를 추가  
Time 시각까지 다운로드 실행 후, 추가

## int removeHub(int Time, int chID)

Time 시각에 chID 허브 삭제  
chID 허브 또는 허브의 자손이 PC와 연결되어 있을 경우 삭제 실패, 0 반환  
그 외의 경우 chID 허브 및 자손 허브 모두 삭제 후 1 반환

## void requestDL(int Time, int parID, int pcID, int Size)

Time 시각에 parID인 서버 또는 허브에 pcID인 PC 추가 후 전송 속도 변경  
Size 크기만큼 데이터 다운 시작  
다운로드 완료된 PC는 자동 접속 해제되며, 즉시 접속 해제, 전송 속도 변경

## Result checkPC(int Time, int pcID)

pcID가 다운로드 중이면 finish = 0, param = 남아 있는 다운로드 크기 반환  
완료된 경우 finish = 1, param = 완료된 시각 반환

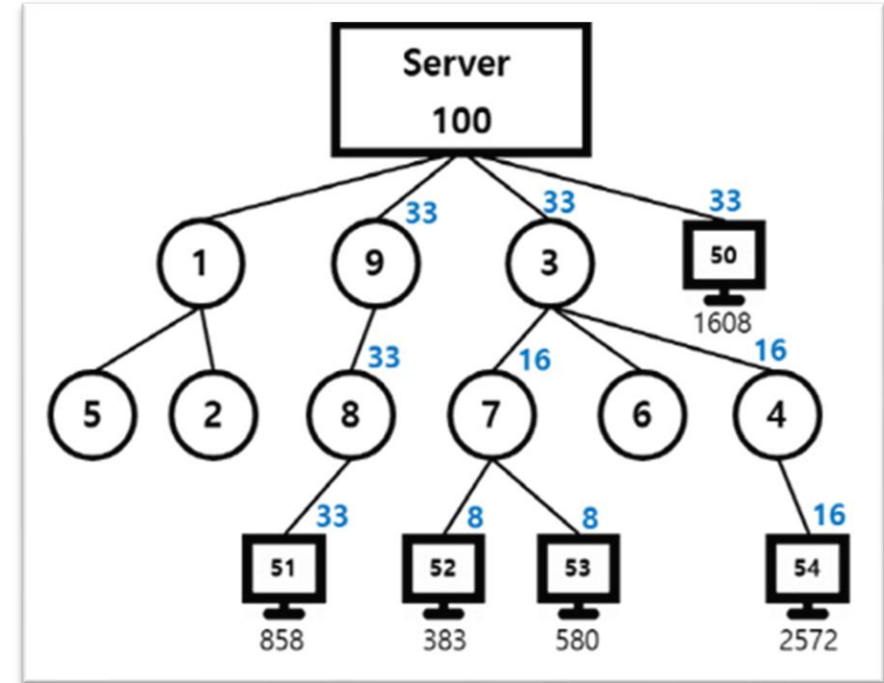
# 문제 분석

## 함수별 수행

- **hub 추가**  
Time까지 다운로드  
트리에 노드 추가
- **hub 삭제**  
Time까지 다운로드  
트리에서 노드 삭제  
삭제 가능 여부 판별 : 하위 노드에 pc가 있는가
- **pc 추가**  
Time까지 다운로드  
트리에 노드 추가, 다운로드 시작(전송속도 업데이트)
- **pc 상태확인**  
다운로드 끝났는지 판별  
끝난 경우, 끝난 시각 반환  
진행중인 경우, 남은 크기 반환

## 주요 포인트

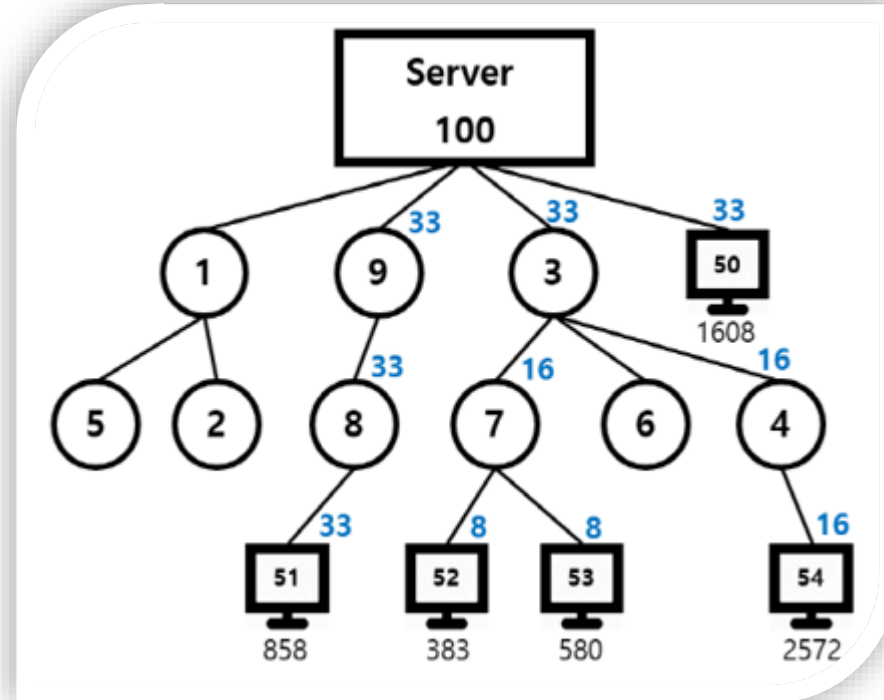
- **트리 구성**  
어떤 정보를 기록할 것인지, 어떤 자료구조를 쓸 것인지
- **timeflow**  
함수 호출 사이에도 다운로드 진행, 함수 처리 전에 Time까지 다운로드
- **전송 속도 업데이트**



# 트리 구성

## 당장 필요해 보이는 정보

- parent
- child list  
자식노드 5개 뿐이므로 array, vector, .. 아무거나 사용
- 남은 데이터양
- 전송 속도
- 전송 종료 시각
- 하위 노드의 총 pc 개수? 전송중인 자식노드 개수?
- 상태 정보(hub/다운로드중인 pc/완료된 pc)?



# Time까지 다운로드 진행

다운로드 진행 시, {전송 속도, 다운로드 완료 시각}이 변경되는 경우

1) pc 추가, 2) pc 전송 완료

방법

time  $\leq$  50,000 , pc개수  $\leq$  1,000

1) current time을 1씩 증가하면서 진행

$O(50,000 * 1,000)$

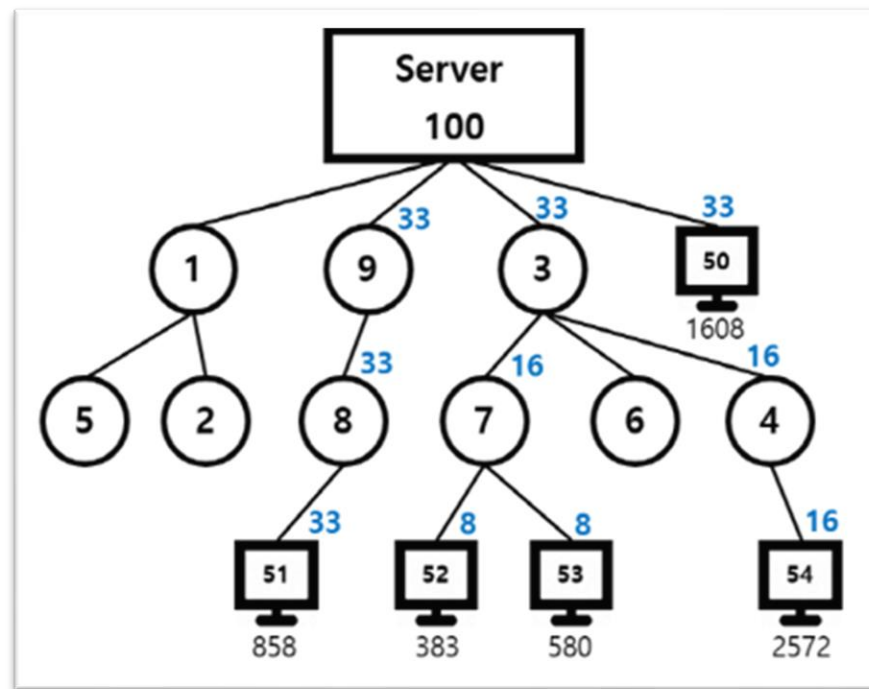
2) 가장 먼저 전송이 종료되는 pc 종료시점을 찾으면서 진행

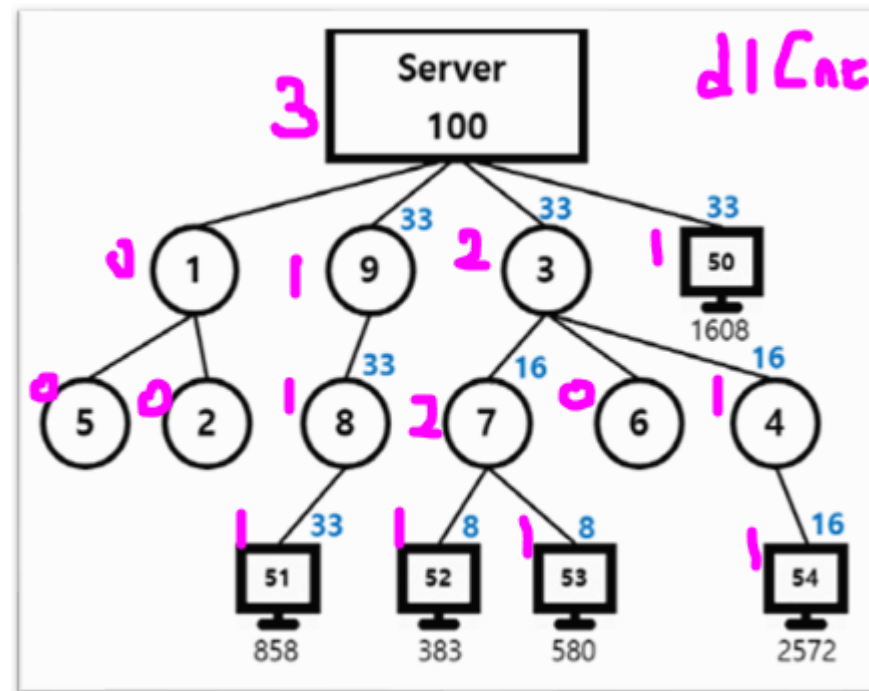
- linear search

$O(1,000 * 1,000)$

- pq / set : {endTime, id}

$O(1,000 \log 1,000)$





# 전송 속도 업데이트 방법

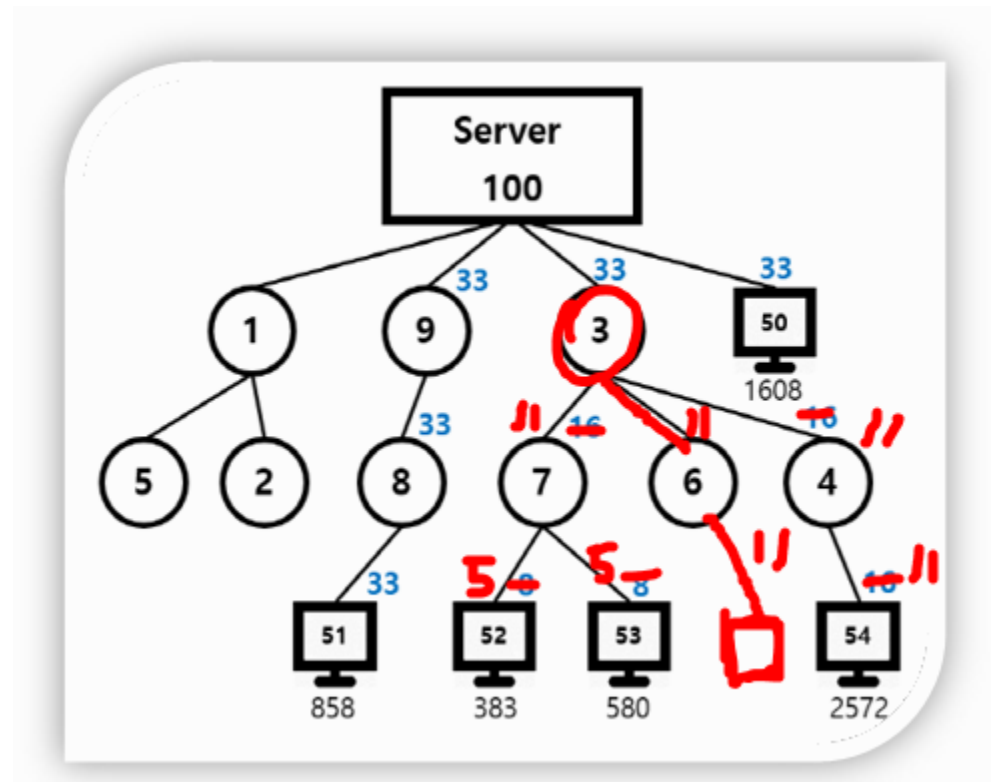
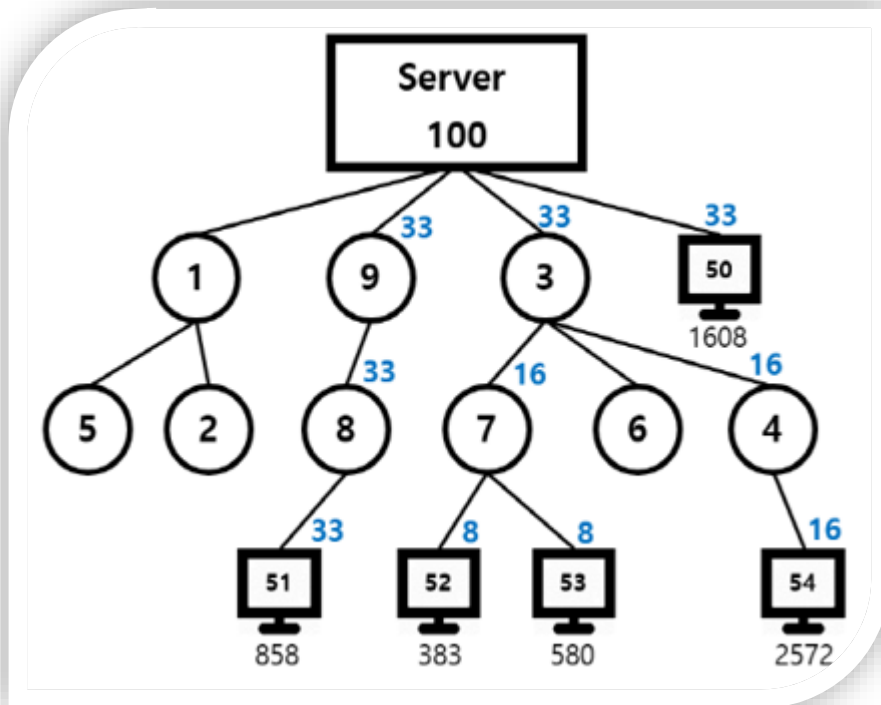
## 1) pc 추가

pc의 조상노드 중 이미 전송중인던 가장 가까운 노드 이하로만 변경된다.

pc에서 dlCnt 개수를 증가시키며 부모노드를 타고 올라간다.

그 과정에서 dlCnt가 2 이상인 노드가 나오면 이미 전송중이었던 노드이므로 그 위부터는 변화 없으며,

해당 노드부터 모든 자손 노드의 전송속도를 및 정보를 변경해나간다.



# 전송 속도 업데이트 방법

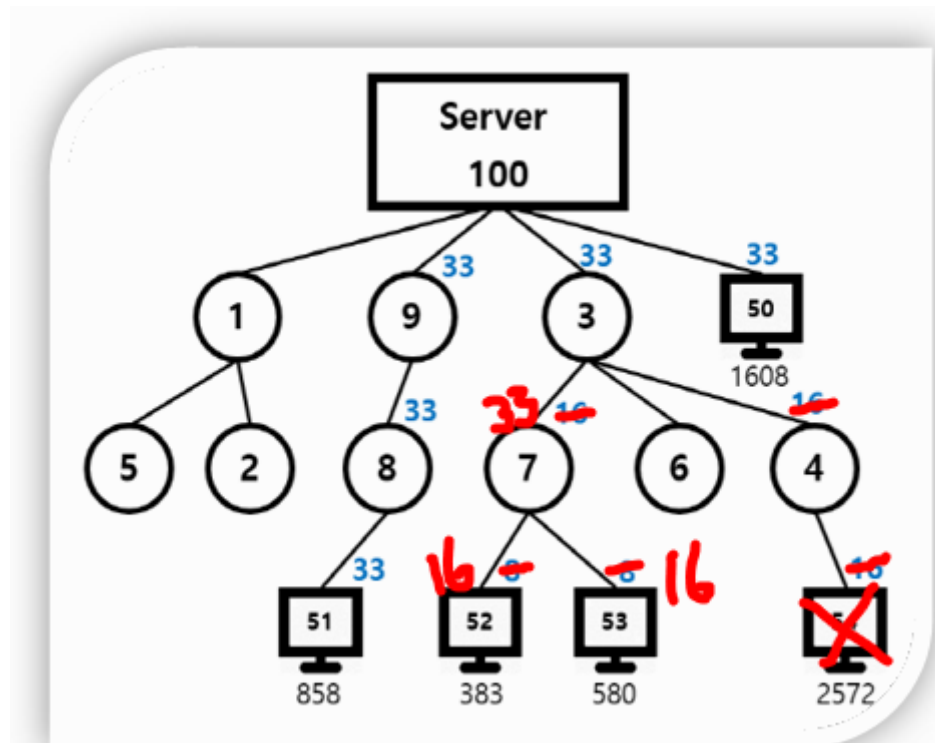
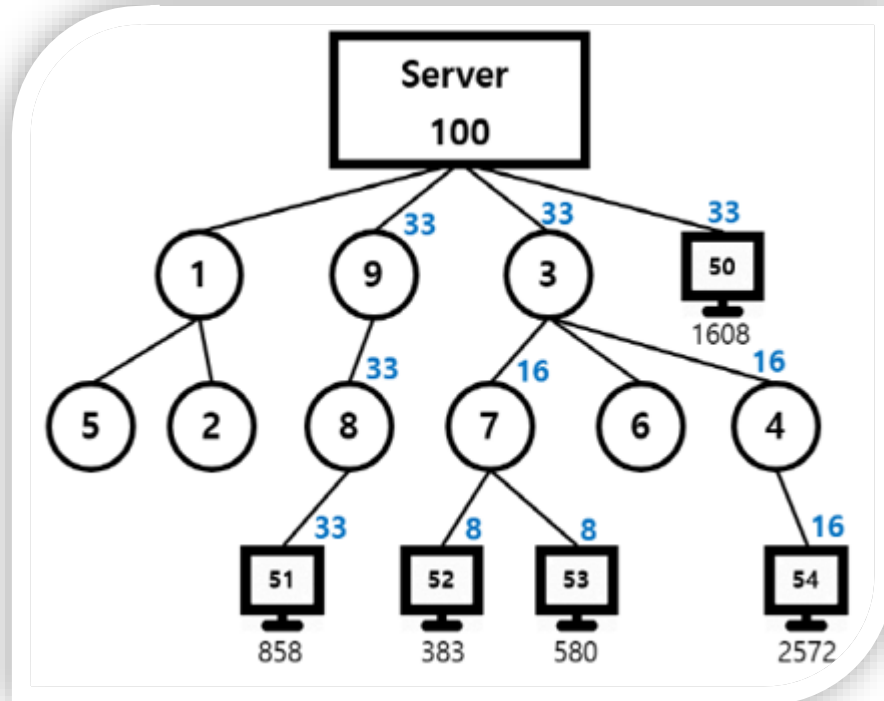
## 2) pc 삭제

pc의 조상노드 중 pc삭제 이후에도 데이터를 전송하는 가장 가까운 노드 이하로만 변경된다.

pc에서 부모노드를 타고 올라가면서 dlCnt 를 감소시킨다.

그 과정에서 감소 후에도 dlCnt가 1이상이면 그 위부터는 변화 없으며,

해당 노드부터 모든 자손 노드의 전송속도 및 정보를 변경해나간다.





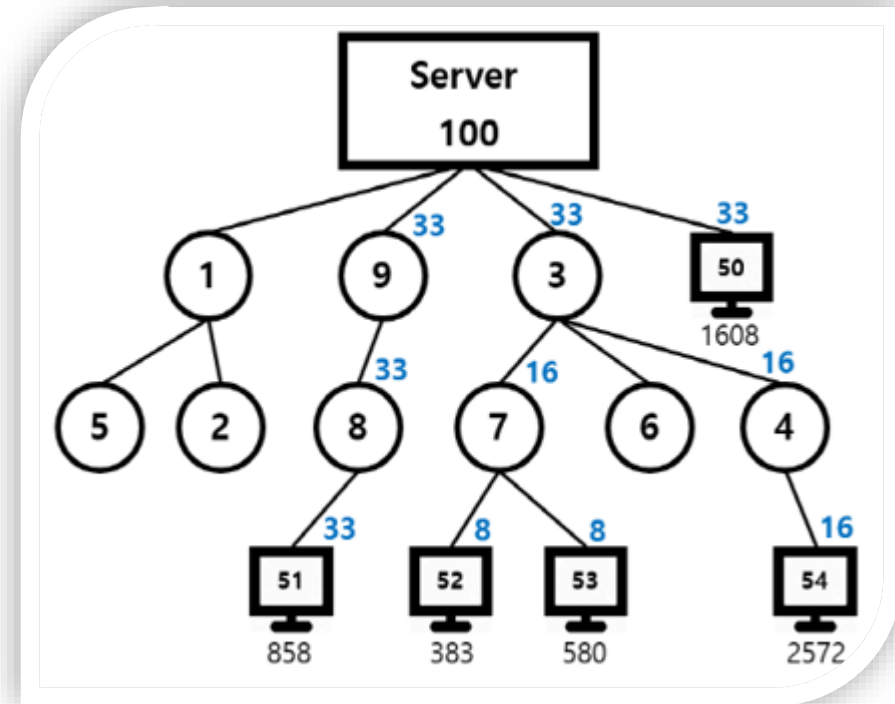
# 최종 형태

## 노드별 필요 정보

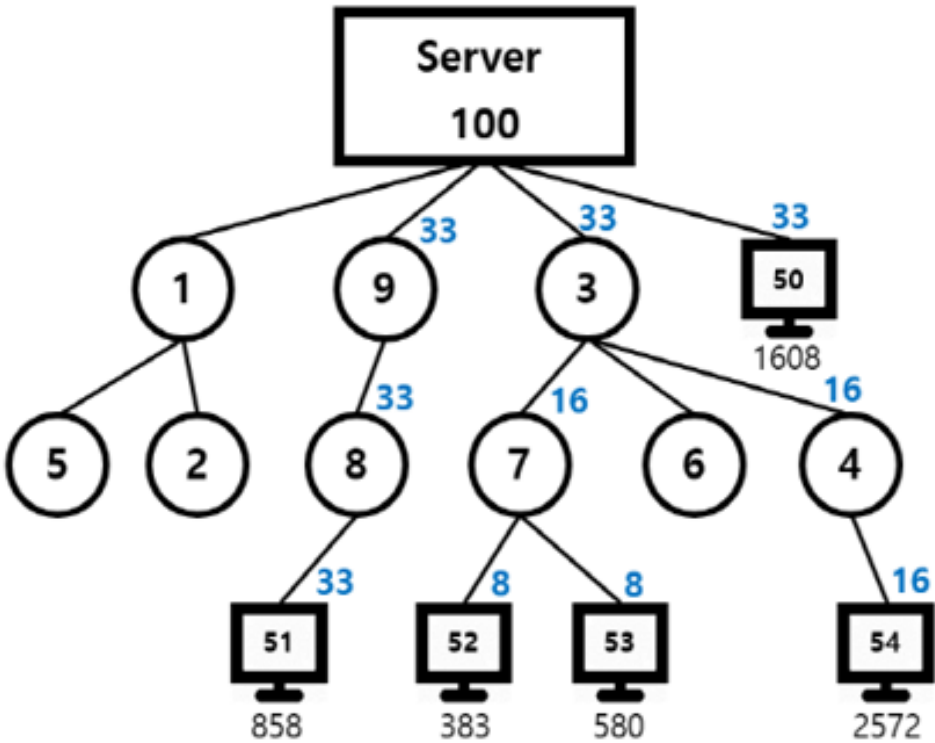
- parent
- child list (array or vector)
- 전송속도, 남은크기
- 시작시각, 종료(예정)시각
- 전송중인 자식노드 개수

먼저 끝나는 순으로 PC 관리 : {endTime, id}

- 1) set
- 2) pq



예제



Order 15. requestDL(30, 4, 54, 2572) 수행 후 상태

Order	Function	Note	Figure
16	removeHub(50, 9)	return 0	
17	removeHub(60, 9)	return 1	
18	checkPC(61, 50)	finish = 0, param = 500	
19	checkPC(63, 51)	finish = 1, param = 56	
20	checkPC(65, 52)	finish = 0, param = 67	
21	addHub(69, 0, 100)		
22	addHub(70, 0, 110)		
23	addHub(71, 0, 120)		
24	requestDL(75, 5, 55, 5432)		[Fig. 3]

ID	speed	dICnt
0	100	3
1	0	0
2	0	0
3	33	2
4	16	1
5	0	0
6	0	0
7	16	2
8	33	1
9	33	1

ID	speed	start	end	size
50	33	30	79	1608
51	33	30	56	858
52	8	30	78	383
53	8	30	103	580
54	16	30	191	2572

※ 의미 있는 수치만 표현  
왼쪽 그림 상태에서 시작

감사합니다

