

하나의 기기와 협력할 때 발생 가능한 예상 이슈와 그에 대한 고찰 정리

201911140 이호준

1. 협력 도중 로컬 프로세싱을 휴식할 것인가?
 - A. Power를 고려하지 않을 경우 휴식으로 얻을 수 있는 이득이 없다고 생각함
2. 협력 기기와의 통신 상태를 판단하는 기준
 - A. 시간당 전달 가능한 프레임의 수나 시간당 전달 가능한 데이터의 크기
 - B. 이러한 정보의 측정 방법은 파일을 다운로드 할 때 측정되는 MB/sec이나 WIFI를 연결할 때 측정되는 네트워크 속도를 측정하는 방법에 대해 조사하면 찾을 수 있을 것이라 생각함
3. 협력이 도움이 된다고 판단하는 기준
 - A. (데이터의 네트워크상 이동에 소요되는 시간 + offloading에 소요되는 시간 < 로컬 프로세싱으로 해당 데이터를 처리하는 시간)의 조건을 만족할 경우 도움이 된다고 판단할 수 있음
 - B. 로컬 프로세싱으로 처리하는 시간은 FPS를 통해 계산할 수 있고, 데이터의 네트워크상 이동에 소요되는 시간은 [2]에서 측정한 수치를 사용할 수 있음
 - C. 갑작스러운 연결 상태 변화를 통해 매우 짧은 시간 동안 [A]의 조건이 충족되지 않는다고 협력을 중단하는 것을 피하기 위해 [A]의 조건이 n초 이상 [A] 조건을 만족하지 않을 시 해당 기기와 협력을 중단할 것인지 적절한 시간 설정이 필요할 듯, 실험 데이터를 통한 결정이 필요
4. 협력 도중 더 나은 협력자를 찾기 위해 탐색 신호를 보내는 조건
 - A. 일정 시간 마다 보내기
 - i. 현재 협력자보다 더 나은 후보를 찾을 가능성 있음
 - B. 특정 조건 만족 시 보내기
 - i. 현재 협력자의 통신 상태가 악화되어 [3-A]의 조건을 만족하지 못할 경우
 1. [2-A]의 조건을 n초 이상 만족하지 않을 때 [4-B]를 실행할 것인지 시간적 조건 추가 필요. [3-C]와 비슷한 맥락
 - C. 협력자의 상태 변화가 잦은 환경이라면 [A]의 주기를 짧게 하여 원활히 대응해야 하고, 단순 실험 환경과 같은 안정적인 환경에서는 [A]의 주기를 길게 하거나 [B] 조건

만을 사용해도 무방할 것이라 생각함

5. 협력 도중 더 나은 통신환경을 가진 협력자가 등장했을 때
 - A. 어떤 수치가 얼마나 차이날 때 협력을 전환할 것인가?
 - i. 어떤 수치가: [2]에서 통신 상태를 판단할 때 사용한 정보
 - ii. 얼마나: 새로운 협력 후보의 [5-A-ii]의 수치가 기존 협력자 몇% 우수한 지
 - B. 새로운 협력자의 양호한 연결 상태로 얻을 수 있는 이득보다 협력을 전환할 때 발생하는 손실이 클 수 있으므로 [5-A-ii]의 값을 설정해야 함
6. 데이터와 결과값을 안정적으로 매칭할 수 있는 구조
 - A. 오류로 인한 결과값 손실, 통신 단절로 인한 결과값 손실 등이 최대한 발생하지 않도록 높은 안정성과 손실 발생을 빠르게 감지하고 쉽게 해결할 수 있도록 유연성을 갖추어야 함
 - B. 해당 파트에 대해서는 추가적인 보완 필요
7. 갑작스러운 통신 단절에 대한 대응
 - A. 단절로 인해 결과값이 손실된 데이터에 대한 처리
 - i. 어떤 데이터의 결과가 손실되었는지 확인할 방법
 - ii. 확인된 결과 손실 데이터를 처리할 방법
 - iii. 단절된 협력을 재개할 방법
8. 협력과 로컬의 전환
 - A. 한 번의 협력 사이클동안 전송할 프레임의 수를 결정
 - i. 통신 연결 상태의 안정성을 고려해야 함
 - ii. 안정성에 관한 자세한 논의는 다음 파트에서 계속
 - B. 안정성이 높다면 많은 프레임을 가끔, 안정성이 낮다면 적은 프레임을 자주 주고받는 것이 유리
9. 안정성의 수치화
 - A. 데이터 전송 속도 최고값과 최저값의 차이
 - B. 평균 전송 속도의 오차범위 $n\%$ 이내로 유지된 시간
 - C. 이 외에도 메인 기기와 협력 기기의 연결 상태의 안정성을 수치화 할 수 있는 수학

적인 방법에 대해 논의가 필요할 듯

10. 로컬과 협력으로 처리하는 데이터의 비율

- A. 얼마나 많은 수의 프레임을 협력으로 맡길 것인지 결정
 - i. 데이터 전송 속도를 고려해야 함
- B. 만약 로컬 프로세싱이 100초(메인 기기와 협력 기기에서 동일), 이동 + offloading 소요시간이 왕복 80초일 때 100프레임을 처리하는 케이스를 가정
 - i. 전부 로컬 프로세싱인 경우 10000초
 - ii. 전부 협력으로 처리하는 경우 18000초
 - iii. 절반을 협력으로 처리하는 경우 9000초 (50프레임을 먼저 보내고 [2000초], 50프레임씩 메인과 협력기기에서 동시에 프로세싱 [5000초], 결과값 받기[2000초], 계산의 편의를 위해 프레임과 결과값의 전달 시간을 같다고 가정)
- C. 협력이 도움이 되는 조건을 만족하는 협력기기라도 [10-B-iii]와 같이 협력 비율을 잘 못 잡으면 오히려 손해가 발생, 최대한의 이득을 얻기 위해서 메인기기와 협력기기에서 “동시에” 프로세싱이 발생하는 시간이 최대한 늘어나도록 조절해야함
- D. 실제 상황에서는 더 복잡한 전환과정이 있을 것이고 이에 대한 최선의 전환 알고리즘을 생각해야함