

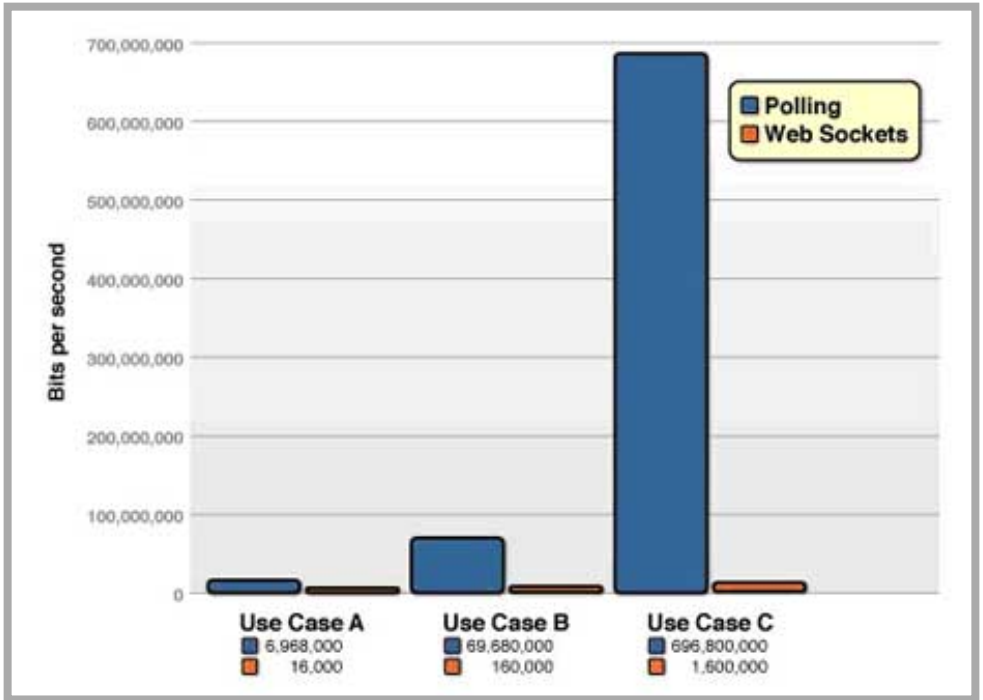
## Socket.IO 사용시 집고 넘어야 할 한 가지

---

웹소켓과 폴링(polling)방식의 소켓이 공존하는 [Socket.IO](#)를 사용해 보면서 간과해서는 안될 점에 대해서 별도의 포스트로 작성합니다. 이전 포스트에 작성한 내용이지만 주제와 무관하여 다시 작성했습니다.

지금까지 여러 웹소켓 테스트를 통해 알아낸 사실들 중 꼭 집고 넘어가야 할 하나는 폴링방식과 웹소켓(플래시소켓 포함)을 무분별하게 사용하면 위험하다는 것입니다. 이것은 브라우저에서 HTTP 요청이 발생하느냐 마느냐의 차이이며 요청이 발생하는 경우 클라이언트나 서버 사이드 모두에 적잖은 오버헤드를 감수해야합니다. 그렇기 때문에 개발하고자하는 애플리케이션이 어떠한 빈도로 통신 횟수가 발생하는 수준에 따라 "폴링 방식 포함" 또는 "웹소켓 방식만 지원"으로 구분해서 취급해야 합니다. 조금 과한 예를 들자면, 초당 20프레임을 확보해야하는 리얼타임 웹앱에서 폴링방식의 소켓을 활성화하는 것은 DDoS 공격을 스스로 구현하는 자살행위나 다름없습니다.

아래의 표는 폴링과 웹소켓을 각각 사용하는 애플리케이션에서 불필요한 네트워크 처리량으로 인한 오버헤드를 비교한 것으로 [Kaazing에서 폴링과 웹소켓을 비교 분석](#)한 자료중의 일부입니다.



Use case A: 1,000 clients receive 1 message per second:  
 Network throughput is  $(2 \times 1,000) = 2,000$  bytes = 16,000 bits per second (0.015 Mbps)

Use case B: 10,000 clients receive 1 message per second:  
 Network throughput is  $(2 \times 10,000) = 20,000$  bytes = 160,000 bits per second (0.153 Mbps)

Use case C: 100,000 clients receive 1 message per second:  
 Network throughput is  $(2 \times 100,000) = 200,000$  bytes = 1,600,000 bits per second (1.526 Mbps)

덧, 로드 밸런싱으로 어찌할만한 수준이 아닙니다.