|  |
| --- |
| **Reading Assignment #2 The Design Philosophy of the DARPA Internet Protocols** |

미국 국방성은 70년대 Packet Switched Network 기반의 프로토콜을 개발하였는데, 여기에는 Internet Protocol(IP), Transmission Control Protocol(TCP)가 포함된다. 프로토콜이 어떤 이유로 현재의 모습을 갖추게 되었는지에 대한 추론은 쉽지 않은 것이 사실이다.해당 논문은 인터넷/프로토콜이 현재와 같은 모습(대표적인 예로 connectionless 혹은 datagram mode)으로 발전되게 되었는지 설명하는 데 주안을 두었다.

**#디자인 시 가장 중요한 목표 :**DARPA 인터넷에서는 현존하는 네트워크들을 효과적으로 다중-이용하는 방법을 개발하는 데 주안점을 두었다. 인터넷을 구분하는 기본은 네트워크로 그 시작은 ARPANET과 ARPA Packet Radio 네트워크를 연결하는 것이었다. 또 다른 방법으로는 그냥 하나의 시스템으로 네트워크를 묶는 방법이 있었는데, 성능 면에서나 결합 측면에서 이점이 많았지만 인터넷이 현존하는 네트워크를 합치는 데 특히 의미가 컸으므로 개별 네트워크를 묶어 인터넷을 구성하는 방법을 사용하게 되었다. Circuit Switching 등 여러 방법이 많은데도 Packet Switching 방식을 이용한 것은 remote login과 같이 Packet Switching을 이미 이용하고 있는 서비스 때문이었다. 또 다른 측면에서 인터넷을 구성하고 있는 것은 Gateway인데 이는 Packet Switching 방식에서 store & forward를 위해 존재하는 계층으로 볼 수 있다. 즉, 인터넷은 Packet Switch에 기반한 통신이고, 다수의 네트워크를 묶은 것인데 그 매개체는 Gateway라는 Packet Store & Forward Unit이라고 할 수 있겠다.

**#제 2의목표 :**네트워크나 Gateway를 잃어도 인터넷 연결은 지속되어야 하고, 다수의 통신 서비스를 지원해야하며, 여러 종류의 네트워크를 수용할 수 있어야하고, 자원 이용에 있어 분산된 관리를 할 수 있어야하고 비용도 절감해야하며 host를 붙이는 low-level의 작업이 복잡하지 않아야하고 계량할 수 있어야 한다. 논문 상에 나와 있는 내용은 그 중요성의 순서대로 나열되어 있으며, 이 순서는 해당 네트워크가 필요로 하는 상황에 따라 달라진다. 즉 이 순서가 바뀐다는 것은 네트워크 전체의 성향도 바뀐다는 의미가 되겠다.

**#네트워크의 생존성(Survivability) :** 네트워크나 Gateway 연결이 실패하더라도 인터넷 연결 자체는 성립되어 있어야 한다. 연결이 끊겼을 경우 User-interface 상에서 다시 연결을 시도한다기보다는 Low-level 에서 조용히 연결을 확신시켜주는 형태를 띈다. 이를 위해서 Lower Layer 상에 존재하는 State Information은 loss로부터 보호될 필요성이 있다.

**#인터넷은 그 특성에 따라 여러 서비스를 제공할 수 있어야 한다 :** 논문 상에서 특별한 예로 TCP와 XNET를 들었는데, XNET는 Debugger의 일종으로 Reliable data를 꼭 수신할 필요는 없다. 따라서 굳이 완벽한 데이터를 넘길 필요가 없는 부분에 대해서는 그 필요성에 따라 독자적인 서비스를 구현하게 된다.

**#다수의 네트워크 종류에 대한 호환성 :** 네트워크 종류는 대단히 다양한데(X.25, Ethernet 등) 이들을 연결하게 될 때 가장 중요한 역할을 했던 것은 Packet혹은 Datagram이였다.

**#(Relatively) Minor Objectives :** Distributed Management의 경우 산재된 네트워크들을 한 회사가 관리하고 있지 않다는 점이 가장 큰 문제가 된다. 그리고 Cost efficiency 측면에서 Client가 보내는 메시지의 길이에 따라 그 효율성이 높아지기도 하고 낮아지기도 해 이 부분은 쉽게 조절하기 힘든 측면이 있다고 설명하고 있다. 또한 retransmission 문제도 효율성 측면에서 꽤 많은 문제를 야기하고 있다. 또 프로토콜 내부를 잘 모르는 프로그래머에게 Host Attachment는 상당한 노력이 수반될 수 있어 이 부분도 효율적이지만은 않다. 특히, Host의 프로그래머가 직접 프로그램을 했을 때 Implementation이 잘못 되었을 경우 전체 시스템에도 영향을 줄 수 있어 이는 중요한 문제이기도 하다. Accountability 문제는 근래에 들어서야 Military-Purpose가 아닌 상업 목적에 의해 고려되기 시작한다.

**#네트워크 디자인 문제 :** Network Architect에게는 네트워크에 관한 Clear Specification이 존재할 수 없다는 게 가장 큰 문제라고 생각된다. (논문 상에 Guidance가 없다고 표현됨) 따라서 고객, 특히 초창기 국방부의 요구를 맞추는데 있어서도 문제가 있었을 수 밖에 없다.

**#Datagram :** 인터넷의 기본 구조는 Datagram에서 시작한다고 생각해도 과언이 아니다. 중간 노드에서 Connection State를 빼버려 재 연결 시 상태에 관한 고려 없이 재 연결을 시도할 수 있도록 하였다. 또한 데이터전송의 최소 단위로 여러 서비스가 구현될 기본 단위를 만들어주었으며 많은 종류의 네트워크를 연결하는 데 있어서도 통신 최소 단위로서의 역할을 하며 네트워크 연결에 지대한 영향을 미쳤다.

**#TCP :**Flow Control에 있어 초창기에는 패킷과 Byte를 이용하였으나, 두 개는 과잉이라는 의견으로 결국 Byte 단위를 쓴다.

논문에 제시된 Priority만을 고려하였을 때, 설명된 사항들을 종합해 보면 현재 인터넷은 꽤 합리적으로 구성된 성공적인 인터넷 구조라고 생각할 수 있다. 하지만 Resource Management 나 Accountability 등 몇 개의 문제에 대해서는 Datagram이 그 근본적인 원인으로 지적되는 만큼 다음 세대에서 더 개선된 인터넷 디자인이 나오지 않을까 생각되며 논문을 마친다.