TCP/IP 大约在 1970 年之后被提出，它起源于 DARPA 的项目开发，并被广泛地用于军事领域和商用领域。这篇论文着重于描述 TCP/IP 形成的原因，从历史的角度详细地介绍其设计哲学。

DARPA 互联网架构的顶级目标为，在现有的互联网络基础上，开发出高效的多路传输技术。进一步地，二级目标详细地阐述了如何达到高效的互联：1 即使在网络和网关失效的情况下，互联网通讯也一定要持续；2 互联网必须支持多种通讯服务；3 互联网架构必须适应多种网络；4 互联网架构必须允许资源的分布式管理；5 互联网架构必须是高性价比的；6 互联网架构必须允许低消耗地进行主机互联；7 互联网架构的资源必须是可统计的。

第一个目标强调了互联网架构的高容错性。为了达到这个目的，正在通讯的状态必须得到一定的保护，具体采取的是命运共享模型而不是复制状态模型，因为其可以防止任意多的中转点失效，并且简单便于工程实现。

第二个目标表明互联网架构应该在传输服务的层面上支持多种服务。TCP 采用可靠的数据流传输，而 IP 采用基于报文的协议并可为基础建立多种服务。

第三个目标指出互联网架构应该能够使用多种军事和商用的网络。互联网架构能达到如此的弹性，是因为它对网络本身能提供的功能只做了最小的要求，即网络能传输包或报文，包必须到一定的大小，网络有某种寻址机制。

对于上述的目标来说，目前互联网架构已经很成功了，并被广泛用于商业和军事环境中。另一方面，更多的精力应该被放到资源统计，管理和操作权限管理上。

Reference

The Design Philosophy of the DARPA Internet Protocols. [1988, David D. Clark]

这篇论文介绍了一项有助于在分布式计算机系统中安排功能模块位置的设计原则。这一原则称为端到端主张，它指出如果在系统的低层次放置功能,那么获得的效果和为此付出的代价相比，可能是价值不大的。本文讨论的例子包括比特错误恢复，安全使用加密，重复消息剔除，系统崩溃恢复和送达确认。

作者对端到端主张的一个应用“谨慎文件传输”做了介绍。文章列出了 5 种基本的威胁，包括：读文件错误；软件拷贝文件错误；硬件拷贝文件错误；多余 1 位改变；任何时间主机宕机。作者列举了更多端到端的应用领域，如传输确认，数据安全，复制信息限制，确保 FIFO 信息传递，交换管理等。我们可以灵活地运用它，为设计协议和网络体系结构进行服务。

谈及在通信子系统如何选择要提供的功能时，端到端主张是某种“奥卡姆剃刀”。由于通信子系统经常会比使用它们的应用程序更早确定，设计人员就可能受到采用更多的功能来“帮助”那些使用该系统的应用程序的诱惑。了解端到端主张，有助于抵御这种诱惑。如今谈论“分层”的通讯协议显得很时髦，但在各层次中分配功能的办法则没有得到明确定义。分层可以提高模块化。端到端主张可以被视为组织此类分类系统的一系列原则中的一部分。作者希望本文的讨论能为有关“正确”分层的争论提供新的思路。

Reference

END-TO-END ARGUMENTS IN SYSTEM DESIGN. [1984, J.H. Saltzer, D.P. Reed and D.D. Clark]

互联网及其架构从其最初的设计到目前的状态已经经历了较大的变化，而不是一层不变。而正是这个变化发展的过程，造就了互联网今天的成功。然而，对今天的互联网架构原则进行剖析还是非常重要的。

互联网社区的很多成员认为互联网没有架构，有的只是 25 年累积下来的传统；在互联网层次上只需要一种协议是最理想的；端到端功能最好被端到端协议实现；没有人掌控互联网，互联网是没有中心的，没有人可以把互联关掉。

异质性是不可避免的；如果有多种方式来解决一个问题，仅仅选择一个；所有的设计必须有良好的可扩展性;性能和消耗也一定要被考虑,就像功能性一样；选择最简单的解决方案；模块性很好，如果能将事情分割开的话，就这样做；采取一个现在接近完成的解决方案，比找到一个完美的解决方案更好；尽可能地避免选择和参数，选择和参数应当能够动态配置都不是手动设定；在发送的时候尽量严格，在接受的时候尽量容错性高；对未经允许的包谨慎处理，特别是多播和广播产生的；环依赖一定要避免掉；对象应当能自我描述；所有的说明应当使用相同的术语和符号；当没有多个实例部署运行的情况下，不要去建立标准。

避免任何设计如果需要被硬编码或者存储在非易失性存储上；应当仅仅使用一个命名结构；公开命名应当使用独立的 ASCII 码；地址一定要没有歧义；上层协议一定要能无歧义地识别出端点。

所有设计必须要适用于 IP 安全体系;我们期望互联网运营商保护所有流量的安全和隐私；选用的加密算法应当被广泛地验证有很强的安全性能；明确跨端点的安全性。

Reference

Architectural Principles of the Internet. [1996, B. Carpenter]

一直以来，鲁棒性是互联网核心设计。对于鲁棒性的工作很多集中于“错误-终止”模型，它指的是节点错误是完整的，并且很容易被其他节点检测出来的。鲁棒性在很大程度上依赖于一些严谨的设计决策，比如在何处初始化恢复，和如何保持状态等等。

然而，互联网仍然对很多任意的错误异常得脆弱，比如说简单的错误配置路由状态，严重地影响互联网的功能。目前，互联网在全球的通讯基础扮演非常重要的角色，这种程度的脆弱性是不可接受的。

这篇论文讨论如何让互联网对于这些任意错误更加的鲁棒性，需要从网络协议的设计上进行改变。为了达到这个目的，论文提出了六点设计指导来提升网络协议的设计。这些指导来源于过去一系列错误的研究，研究解决了如何在第一阶段防止这些错误的问题。这些指导背后的统一主题是，我们需要设计出更具防御性的网络协议，需要考虑到每一个角落的恶意攻击，错误实现和错误配置。

具体地：1 价值概念简化性：使用非常简单的接口，拥有明确的功能语义，不在协议定义中嵌入性能优化；2 最小化依赖性：如果其他节点潜在得不可信，协议需要显式地设计用来减少相互依赖；3 尽可能地验证：不可能消除所有的依赖项，尽量验证来自其他节点的信息；4 保护资源：为了防止来自未验证第三方的未经许可的请求；5 限制脆弱性的范围：因为错误总是不能被全部防止或发现，我们在设计的时候需要限制错误造成的影响；6 公开错误：我们需要在错误叠加之前进行发现和更正。

Reference

Design Guidelines for Robust Internet Protocols. [2003, Tom Anderson, Scott Shenker, Ion Stoica and David Wetherall]