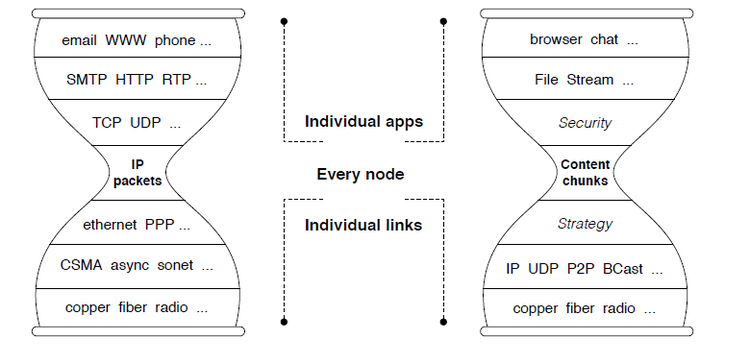
命名数据网络

# 背景

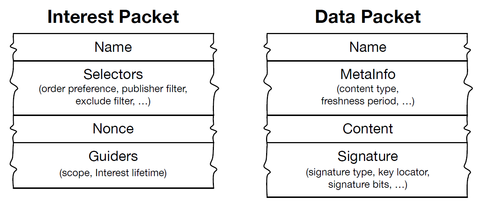
当今互联网的沙漏体系结构以通用网络层IP为中心，IP层实现了全球互连所需的最小功能。当代互联网体系结构围绕基于主机的对话模型展开，该对话模型创建于20世纪70年代，允许分布在不同地理位置的用户使用几台大型，不移动的计算机。[5]这种瘦腰使互联网的爆炸性增长能够让上层和下层技术独立创新。但是，IP旨在创建一个通信网络，其中数据包仅命名通信端点。电子商务、数字媒体、社交网络和智能手机应用的持续增长导致了互联网作为分销网络的主导性使用，分布是网络比通信网络更加普遍，通过点对点通信协议解决分布问题是复杂且容易出错的。

命名数据网络（NDN）项目提出了IP体系结构的演变，该体系结构概括了瘦腰的作用，使得数据包可以命名通信终端以外的对象。更具体地说，NDN将网络服务的语义从传递数据包改变为给定的目标地址，以获取由给定名称标识的数据。NDN数据包中的名称可以命名任何内容 - 端点，电影或书籍中的数据块，打开某些灯光的命令等等。希望这种简单的更改允许NDN网络应用几乎所有的互联网的经过良好测试的工程特性可应对端到端通信以外的更广泛问题。[[6]](https://en.wikipedia.org/wiki/Named_data_networking" \l "cite_note-6) NDN应用30年网络工程经验教训的例子是网络流量的自我调节（通过兴趣（数据请求）和数据包之间的流量平衡）和安全原语（通过所有命名数据上的签名）被集成到协议中从头开始。



# 体系结构

 NDN中的通信由接收者（例如数据消费者）通过交换两种类型的包驱动，兴趣包和数据包。两种类型的包承载了标识数据片的名字，该数据片可以在包中传输。消费者把期望数据片的名字放进兴趣包，并将其发送到整个网络。路由器使用该名字向数据生产者转发兴趣包。一旦兴趣包到达拥有请求数据的节点，节点将返回包含名字和内容的数据包，伴随被生产者密钥绑定了名字和内容的签名。数据包沿着兴趣包到达的相反路径返回请求的消费者。



当感兴趣的数据包到达时，NDN路由器首先检查内容存储库是否匹配数据; 如果它存在于路由器中，则返回感兴趣的接口上的数据包。否则，路由器在其PIT中查找该名称，并且如果存在匹配条目，则它简单地将该兴趣的传入接口记录在PIT条目中。在没有匹配的PIT条目的情况下，路由器将根据FIB中的信息以及路由器的自适应转发策略将兴趣转发给数据生产者。当路由器从多个下游节点接收到相同名称的兴趣时，它只向上游转发数据生产者。

当数据包到达时，NDN路由器找到匹配的PIT条目并将数据转发到该PIT条目中列出的所有下游接口。然后，它删除该PIT条目，并在Content Store中缓存数据。数据包总是采用兴趣的反向路径，并且在没有包丢失的情况下，一个兴趣包在每个链路上产生一个数据包，从而提供流量平衡。为了获取包含多个数据包的大型内容对象，兴趣在控制流量流量方面发挥着与当今互联网中的TCP ACK类似的作用：由数据消费者控制的细粒度反馈环路。

兴趣和数据包都不包含任何主机或接口地址; 路由器根据数据包中携带的名称向数据生产者转发兴趣数据包，并根据Interests在每一跳设置的PIT状态信息将数据包转发给消费者。这种兴趣/数据包交换对称性引入逐跳控制环路（不要与对称路由或路由混淆！），并且不需要源或目的节点在数据传输中的任何概念，这与在IP的端到端数据包交付模式中。

为了实现兴趣包和数据包转发功能，每个NDN路由器维护三个数据结构：待定兴趣表（PIT），转发信息表（FIB）和内容缓存（CS）（见图3），和转发策略模块（没有显示在图中），以决定是否、何时、何地转发每个兴趣包。PIT存储了路由器转发但还没有被满足的所有兴趣包。每个PIT条目记录了兴趣包中承载的数据名，以及进接口和出接口。当兴趣包到达，NDN路由器首先检查内容缓存是否有匹配数据，如果存在匹配数据，路由器在兴趣包到来的接口上返回数据包。否则，路由器在PIT中查询名字，如果匹配的条目存在，该条目简单地记录了兴趣包进接口。如果不存在匹配的PIT条目，路由器将基于FIB中的信息和路由器中的自适应转发策略向数据生产者转发兴趣包。当路由器从多个下游节点收到相同名字的兴趣包，它仅仅向数据生产者上游转发第一个兴趣包。FIB自身由基于名字前缀的路由协议填充，每个前缀拥有多个出接口。

Content Store

Content Store

Pending Interest Table

FIB

Pending Interest Table(PIT)

Add incoming interface

Add incoming interface

Data

Interest

Forward

Downstream

forward

cache

Discard Data

Upstream

**×** lookmiss

**√** Lookup hit

**×**

**×**

**×**

**×**

√

√

√

√

Data

转发策略可能在某种特定情形下决定扔掉兴趣包，例如，如果所有的上游链路都是拥塞的，或者兴趣包被用作DoS攻击的一部分。对每个兴趣包，转发策略从FIB中获取最长前缀匹配条目，并决定何时何地转发兴趣包。内容缓存是路由器接收到的数据包的临时缓存。因为NDN数据包是位置独立的，它可以被缓存来满足未来的兴趣包。

       当数据包到达时，NDN路由器找到匹配的PIT条目并向PIT条目中列出的所有的下游接口转发数据。然后路由器移除PIT条目并在内容缓存中缓存数据包。数据包总是沿着兴趣包到达的相反路径传送，并且不存在包丢失情况，在每条链路上，一个兴趣包将返回一个数据包，提供流平衡机制。为了获取大的由多个包组成的内容对象，兴趣包在控制流量方面提供了一个与今天因特网的TCP ACK相似的角色：被数据消费者控制的细粒度的反馈环路（见2.1节）。兴趣包和数据包都不携带任何主机或接口地址；路由器基于包中携带的名字向数据提供者转发兴趣包，并基于兴趣包在每跳建立起来的PIT状态信息向消费者转发数据包。该兴趣包/数据包交换的对称性减少了逐跳控制环路（不要与对称路由或者路由混淆），并消除了在数据交付过程中源和目的节点任何概念上的需要，而不像在IP的端到端包交付模型中那样。

# 命名

## 设计

尽管路由器识别名字中组件之间的边界，他们不将意义归属到名字，NDN名字对于网络是不透明的。这个设计决定允许每个应用程序选择适合他需要的命名方案，因此，命名能够随网络独立地进化。NDN设计假定层次化结构化的名字，例如，被UCLA生产的的视频可能拥有名字/ucla/videos/demo.mpg,‘/’以文本表示法描绘了名字组件，与URL相似。层次化结构允许应用程序表示数据元素的内容和关系。例如，例子中UCLA视频的第3段版本1可能命名为/ucla/videos/demo.mpg/1/3。层次化结构也允许名字汇聚，例如，/ucla能够与产生视频的自治系统通信。扁平名字可以作为特殊案例被兼容，这在本地环境可能是有用的，然而，层次化命名空间在扩展路由系统和提供必需的数据上下文环境方面是最基本的。（甚至提倡扁平路由确认，扁平名字通过引入层次是可扩展的。）

## 结构

NDN设计采用分层结构的名称，例如，由UCLA制作的视频名称可能是/ucla/videos/demo.mpg，其中'/'描述文本表示中的名称组件，类似于URL。这种等级结构有许多潜在的好处：

#### 关系规范：允许应用程序表示数据元素的上下文和关系。EX：加州大学洛杉矶分校演示视频版本1的第3部分可能被命名为/ucla/videos/demo.mpg/1/3。

#### 名称聚合：/ ucla可以对应于始发视频的自治系统

#### 路由：允许系统扩展并帮助为数据提供必要的上下文

## 指定名称

为了检索动态生成的数据，消费者必须能够确定性地为所需的数据构建名称，而无需通过以下方式查看名称或数据：

#### 一种算法允许生产者和消费者根据两者都可用的信息得到相同的名字

#### 兴趣选择器与最长前缀匹配一起通过一次或多次迭代获取所需数据。

目前的研究正在探索应用程序应该如何选择可以促进应用程序开发和网络传输的名称。这项工作的目的是开发和完善现有的命名原则和指导原则，将这些规则转化为系统库中实现的命名约定，以简化未来的应用程序开发

## 命名空间

可以全局检索的数据必须具有全局唯一的名称，但用于本地通信的名称可能只需要本地路由（或本地广播）来查找匹配的数据。单个数据名称在各种范围和背景下都可能有意义，从“这个房间的灯光开关”到“世界上所有的国家名称”。命名空间管理不是NDN体系结构的一部分，就像地址空间管理不是IP体系结构的一部分一样。但是命名是NDN应用程序设计中最重要的部分。使应用程序开发人员（有时候用户）能够设计自己的名称空间以进行数据交换具有以下几点好处：

#### 增加了应用程序数据与其网络使用之间的密切关系

#### 减少对二级表示法的需求（将应用配置映射到网络配置的记录保存）

#### 扩大了开发者可用的抽象范围。

# 路由

## 地址

NDN根据名称路由和转发数据包，消除了IP体系结构中地址造成的三个问题：

#### 地址空间耗尽：NDN命名空间是无界的

#### NAT穿越：NDN消除了公共或私人地址，因此不需要NAT

#### 地址管理：本地网络不再需要地址分配和管理。

## 协议

NDN可以使用传统的路由算法，如链路状态和距离矢量。NDN路由器不公布IP前缀，而是宣布名称前缀，其中包含路由器愿意服务的数据。传统的路由协议，例如OSPF和BGP，可以用来路由名称前缀，方法是将名称视为一系列不透明的组件，并对Interest数据包中的名称与FIB表进行组件最长的前缀匹配。

## PIT状态

每个路由器的PIT状态支持通过NDN的数据平面进行转发，记录每个未决兴趣和入局接口，并在收到匹配数据或发生超时后移除兴趣。每跳数据包状态不同于IP的无状态数据平面。基于FIB中的信息和性能测量结果，每个路由器中的自适应转发策略模块可以做出明智的决策：

#### 控制流量：由于每个兴趣最多检索一个数据包，因此路由器可以通过控制它保留的未决兴趣的数量来直接控制流量。

#### 多点传送数据传送：记录相同数据到达的接口集合的PIT自然支持该功能。

#### 更新路径以适应其网络视图的变化。[13]

#### 交付：路由器可以推断哪些兴趣转发给哪些接口，在PIT中允许多少不满意的兴趣以及不同兴趣的相对优先级。

## 兴趣

如果路由器决定不能满足兴趣，例如上游链路断开，FIB中没有转发条目或发生极度拥塞，则路由器可以向发送兴趣的其下游邻居发送NACK 。这种否定确认（NACK）可以触发接收路由器将兴趣转发到其他接口以探索替代路径。PIT状态使路由器能够识别和丢弃循环数据包，从而允许他们自由地使用多条路径来访问同一个数据生产者。数据包不能在NDN中循环，这意味着无需在IP和相关协议中实施生存时间和其他措施来解决这些问题。

# 网络内存储

因为每个NDN数据包携带一个名字和签名，因此它是有意义的，独立于谁请求的和从哪里获取的。因此，路由器可以在内容缓存中缓存接收到的数据包，并使用其满足未来的请求。内容缓存与在IP路由器中的缓冲区内存类似，但是IP路由器在将包转发到目的地后不能重新使用包，而NDN路由器可以。NDN按照数据获取同等地对待存储和网络信道。对于静态文件，NDN获得几乎最佳的数据交付。甚至在多播情况下，动态内容能够从缓存中获益（例如，实时电话会议）或者包丢失后的重传。

除内容缓存之外，体系结构现在支持一个更加持续的大容量的网络内存储，叫做仓库（缩写为Repo）。这种类型的存储能够支持与今天的内容分发网络（CDNs）相似的服务，不是必须设计它以使用创造性的协议技巧作为应用层覆盖（例如，DNS操作）来使得他们工作。

缓存命名的数据增加了不同于IP的对于隐私的担心。在IP中，我们可以检查包头部，可能是有效负载，来得知谁正在消费数据。在NDN网络中命名和缓存数据可能帮助观察什么数据被请求，但是没有目的地址很难确认谁正在请求他（除非我们直接连接到与请求主机相同的子网）。因此NDN提供了一个与当前IP网络根本上不同的隐私保护排序。

一些研究者特别强调网络内缓存作为ICN体系结构的基本获得物。尽管NDN相较于TCP/IP能够支持更多强大的CDN体系结构，NDN也提供许多其他的功能（数据安全，流平衡，独立带来大量利润的状态数据面板），这些功能拥有更多有意义的重要的优势。

# 传输层功能

NDN体系结构没有一个单独的传输层，他移动今天的传输层协议的功能（多路分解、可靠交付和拥塞控制）到应用层，支持库和转发面板的策略模块。传输层信息，比如端口和序列号不是必需的。所有需要传送的信息都在数据包名字中。例如，名字/ucla/videos/demo.mpg/1/3指定了向哪里转发兴趣包（名字/ucla/），哪个应用程序应该接收它们（/video/），任何应用程序指定的信息（版本1段3）。

当应用程序需要可靠交付，应用程序自身或者它的支持库将监视显著的兴趣包的状态，并当需要时重传它们，例如超时之后。NDN的流平衡需求，节点通过在每一跳限制待定兴趣包的数量控制它们自己流量负载的能力，能够在整个网络内提供高效的拥塞控制。如果拥塞损失发生，由于重传兴趣包可以被缓存的数据包在包丢失之前满足，缓存缓解了该影响。因此，NDN可以避免这种发生在今天的因特网中的拥塞崩溃，当包靠近目的地时丢失，从源主机发出的不断的重传，这将消耗许多带宽。

# 安全

## 概述

与TCP / IP相比，NDN将安全责任（或缺乏安全责任）赋予端点，NDN通过要求数据生成者对每个数据包进行加密签名来保护数据本身。发布者的签名确保了完整性，并能够确定数据来源，从而使消费者对数据的信任与获取数据的方式或位置脱钩。NDN还支持细粒度信任，允许消费者推断公钥所有者是否是特定上下文中特定数据的可接受发布者。第二个主要研究重点是设计和开发可用机制来管理用户信任。已经研究了两种不同类型的信任模型：

#### 层次信任模型：密钥命名空间授权使用密钥。携带公钥的数据包实际上是一个证书，因为它由第三方签名，并且该公钥用于签署特定数据[14]

#### 信任网：在不需要事先约定的信任锚的情况下启用安全通信。[15]

## 应用安全

NDN的以数据为中心的安全性对内容访问控制和基础架构安全具有自然的应用。应用程序可以使用相同的命名基础结构对数据进行加密并将其作为命名数据包进行分发，从而有效地将数据安全边界限制在单个应用程序的上下文中。为了验证数据包的签名，应用程序可以像在其他任何内容中那样获取在数据包的密钥定位器字段中标识的相应密钥。但是，信任管理，即如何确定给定应用中特定分组的特定密钥的真实性，是一项主要的研究挑战。与实验方法一致，NDN信任管理研究由应用开发和使用驱动：首先解决具体问题，然后确定常见模式。例如，NLSR的安全需求需要开发一个简单的层次信任模型，使用较低（较接近根）级别的密钥，用于签署更高级别的密钥，其中密钥的发布使用反映其信任关系的名称。在此信任模型中，名称空间与信任委派的层次结构匹配，即/ root / site / operator / router / process。在层次结构中发布具有特定名称的密钥授权它们对特定数据包进行签名并限制其范围。这种范例可以很容易地扩展到其他真实世界信任倾向于遵循层次模式的应用程序，比如我们的建筑管理系统（BMS）用于签署更高级别的密钥，其中发布的密钥的名称反映了其信任关系。在此信任模型中，名称空间与信任委派的层次结构匹配，即/ root / site / operator / router / process。在层次结构中发布具有特定名称的密钥授权它们对特定数据包进行签名并限制其范围。这种范例可以很容易地扩展到其他真实世界信任倾向于遵循层次模式的应用程序，比如我们的建筑管理系统（BMS）用于签署更高级别的密钥，其中发布的密钥的名称反映了其信任关系。在此信任模型中，名称空间与信任委派的层次结构匹配，即/ root / site / operator / router / process。在层次结构中发布具有特定名称的密钥授权它们对特定数据包进行签名并限制其范围。这种范例可以很容易地扩展到其他真实世界信任倾向于遵循层次模式的应用程序，比如我们的建筑管理系统（BMS）[16]由于NDN在每个应用程序的控制下离开信任模型，因此也可以表达更灵活和表达的信任关系。ChronoChat就是一个这样的例子， [15]这激发了对信任网络模型的实验。安全模型是，当前的聊天室参与者可以通过签署新人的密钥来引入新人。未来的应用程序将实施交叉验证模型（SDSI）[13，3]，它提供了更多的验证冗余，允许数据和密钥名称独立，更容易适应各种真实世界的信任关系。

## 路由安全性

此外，NDN像所有NDN数据一样处理网络路由和控制消息，需要签名。这为确保路由协议免受攻击提供了坚实的基础，例如欺骗和篡改。NDN使用多路径转发以及自适应转发策略模块可以减轻前缀劫持的风险，因为路由器可以检测到劫持造成的异常并通过备用路径检索数据。[17]由于NDN分组的参考内容而不是设备，它是棘手恶意靶向特定设备，尽管缓解机制将需要对其他NDN特异性攻击，例如，兴趣洪水拒绝服务攻击。，[18] [19]此外，如果有一个待决兴趣表（保持关于过去请求的状态），可以做出明智的关于如何处理兴趣的前瞻性决定，则具有许多安全优势：[20]

#### 负载平衡：PIT条目的数量是路由器负载的指标; 限制其大小限制了DDoS攻击的影响。

#### 兴趣超时：PIT入口超时提供相对便宜的攻击检测，并且每个PIT条目中的到达接口信息可以支持推回方案，其中下游路由器被告知未服务的兴趣，这帮助检测攻击。