Network System Design

CS6100

Tutorial 02

Named Data Networking (NDN)

Student Details

Name: R Abinav
Roll No: ME23B1004

Date: January 21, 2026

1 What is Named Data Network?

Named Data Networking is a proposed new Internet architecture that focuses on what data you want rather than where it's located. Think of it like this: today's Internet works like the postal system - you need to know exactly where someone lives (their IP address) to send them a letter. NDN works more like asking "Hey, does anyone have the latest cat video?" and the network finds it for you, whether it's on the original server, cached nearby, or anywhere in between.

Instead of securing the connection between computers like the current Internet does, NDN secures the actual content itself. Each piece of data is signed by whoever created it, so you can trust the information regardless of where you got it from.

2 What is the need for NDN?

The current Internet has a fundamental mismatch problem. Most of what we do online - watching YouTube, Netflix, downloading files - is about getting content, not talking to specific computers. But the Internet was designed for computer-to-computer conversations.

Here's why NDN makes sense:

- **Mobile devices everywhere:** Your phone constantly moves between networks, but IP addresses are tied to locations. NDN doesn't care where you are - it just gets you the data.
- **Content is king:** Services like YouTube, Netflix, and Amazon account for more than half of the world's internet traffic, but running them over IP is inefficient and insecure.
- **Built-in security:** Instead of bolting on security features after the fact, NDN bakes it right into how data moves around.
- **Better for caching:** Popular content can be stored closer to users automatically, making everything faster.

3 How NDN Works?

NDN uses two types of packets to communicate:

3.1 Interest Packets (the request)

When you want something, like a video, you send out an Interest packet with the content's name - something like `/ndn/youtube/cat-video/segment1`.

3.2 Data Packets (the response)

The network finds the data and sends it back. Each Interest packet fetches exactly one Data packet back.

3.3 Three Important Tables

Here's the clever part - NDN routers maintain three important tables:

1. **Content Store (CS):** A cache of recently-used data. If you ask for something that just passed through this router, it can give it to you immediately.
2. **Pending Interest Table (PIT):** Keeps track of who asked for what, so when data comes back, it knows where to send it.
3. **Forwarding Information Base (FIB):** Like a directory that tells the router which direction to send your request.

The process is simple: you send an Interest, routers check their cache first, and if they don't have it, they forward your request toward the source. When the Data comes back, it follows the breadcrumbs left by your Interest packet.

4 How Lookup is performed in NDN?

When an Interest packet arrives at an NDN router, the forwarding process follows a specific sequence:

1. **Check the Content Store first:** The router looks for an exact character-by-character match of the name. If found, boom - instant response.
2. **Check the Pending Interest Table:** If someone else just asked for the same thing, the router doesn't send a duplicate request. It just adds your request to the list.
3. **Look up in the FIB:** This is where it gets interesting. The FIB performs a longest prefix match - finding the most specific route that matches your request. It's like searching for `/iiitdm/cs/course101/lecture5` and finding routes for `/iiitdm/cs` or `/iiitdm/cs/course101`.
4. **Forward the Interest:** The router sends your request toward the data source based on the FIB entry.

When Data comes back, the router checks the PIT to see who was waiting for it, forwards it to all those requesters, deletes the PIT entry, and optionally caches the data in its Content Store.

5 What is the computational complexity of NDN Lookup?

This is where things get tricky. NDN forwarding performs lookups on packet names which have variable and unbounded lengths, increasing the lookup complexity compared to IP.

5.1 The Challenges

- **Variable-length names:** NDN names are URL-like hierarchical structures with variable and unbounded lengths, making lookup more complex than fixed-length IP addresses.
- **Multiple table lookups:** For each packet, NDN potentially needs to search three tables (CS, PIT, FIB), though optimizations exist.
- **Longest prefix matching:** A straightforward lookup consumes time proportional to both the number of FIB entries and the length of name prefixes.

5.2 Proposed Solutions

Various solutions have been proposed:

- **Hash tables with binary search:** Can achieve fast lookups but require careful design
- **Trie-based structures:** Better for hierarchical names but can use more memory
- **Bloom filters:** Space-efficient but have false positive rates

Research has demonstrated achieving 10 Gbps forwarding throughput with 256-byte packets and one billion forwarding rules, showing it's technically feasible but requires sophisticated data structures.

6 Compare NDN Lookup with IP Lookup and Layer 2 MAC lookup

6.1 IP Lookup (Layer 3)

- **Address format:** Fixed 32-bit (IPv4) or 128-bit (IPv6) addresses
- **Lookup method:** Routers use longest prefix match on IP routing tables
- **Complexity:** Well-established algorithms, relatively fast
- **Table size:** Growing but manageable (hundreds of thousands of routes)

6.2 MAC Lookup (Layer 2)

- **Address format:** Fixed 48-bit MAC addresses
- **Lookup method:** Switches perform exact match lookups - one-to-one matching of MAC addresses to ports
- **Complexity:** Simplest - just exact matching using hash tables
- **Table size:** Limited to devices on the local network
- **Learning:** Switches dynamically learn MAC addresses by examining the source MAC of incoming frames

6.3 NDN Lookup

- **Address format:** Variable-length hierarchical names (unbounded)
- **Lookup method:** Longest prefix match on hierarchical names
- **Complexity:** Most complex - needs to handle variable lengths and hierarchy
- **Table size:** Potentially much larger due to unbounded namespace
- **Multiple lookups:** Checks three tables (CS, PIT, FIB) per packet

6.4 Simple Analogy

In simple terms: MAC lookup is like finding an exact phone number in a list. IP lookup is like finding the best matching area code. NDN lookup is like finding the longest matching file path in a directory structure - and the paths can be any length.

7 List of Challenges in NDN

Based on research, here are the major challenges:

7.1 Scalability Issues

- Maintaining scalability of the global routing system becomes a challenge since NDN routes using application data names directly instead of IP addresses
- FIB tables can grow explosively, making them unable to fit in existing router memory

7.2 Security Concerns

- Interest flooding attacks - attackers can overwhelm routers with bogus requests
- Cache poisoning attacks can corrupt stored content
- Privacy issues - content names might reveal too much about what users are accessing
- Every packet must be signed and verified, requiring high efficiency in signature generation and verification

7.3 Performance Challenges

- Memory access latency and the performance of in-network caching
- Signature verification adds computational costs during data retrieval
- Fast lookup algorithms needed for variable-length names

7.4 Trust and Key Management

- Need for secure, resilient, and scalable support for public key distribution since all data must be signed
- Designing effective trust models

7.5 Naming Challenges

- Namespace design - how applications should best select data names to facilitate both application development and network data delivery
- No agreed-upon standard for naming conventions yet

7.6 Deployment Hurdles

- Lack of application support is probably the biggest obstacle to NDN deployment
- Need to work alongside existing IP infrastructure
- Caching poses storage challenges in resource-constrained mobile nodes and IoT devices

7.7 Congestion Control

- Controlling congestion in a multi-path, multi-producer, multi-consumer environment

7.8 Technical Complexity

- Routing protocol design
- Content protection and access control
- Mobility support
- Interoperability with current Internet