

Chapter 4

Network Layer: Services and Protocols

নেটওয়ার্ক লেয়ার (Network Layer) হল ইন্টারনেটের তৃতীয় স্তর,
যার প্রধান কাজ হলো ডেটাকে এক হোস্ট থেকে অন্য হোস্টে (end-to-end) পৌঁছে দেওয়া।
এটি Transport Layer-এর segment-কে গ্রহণ করে এবং সেটিকে IP datagram আকারে পাঠায়।

Transport Segment Transmission: From Sender to Receiver

◊ Sender Side

① Transport Layer (TCP/UDP)

- Application data-কে ছোট ছোট segments আকারে ভাগ করে।
- প্রতিটি segment-এ transport header যোগ করে (port numbers, checksum ইত্যাদি)।

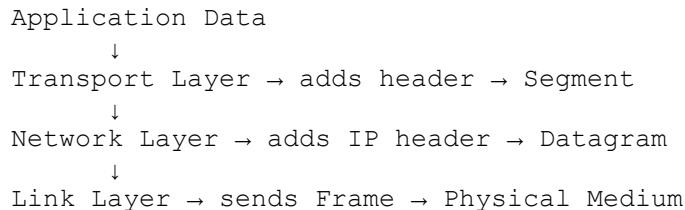
② Network Layer (IP)

- Transport segment-কে encapsulate করে IP datagram তৈরি করে।
- প্রতিটি datagram-এ IP header যোগ করা হয় (source IP, destination IP, TTL, ইত্যাদি)।

③ Link Layer (Ethernet/Wi-Fi)

- Datagram-কে ফ্রেমে পরিণত করে ফিজিক্যাল মিডিয়ায় পাঠানো হয়।

Sender Flow:



◊ Routers (Intermediate Devices)

Routers কাজ করে শুধুমাত্র Network Layer-এ।

- তারা প্রতিটি IP datagram-এর header fields পরীক্ষা করে, বিশেষ করে destination IP address।
- Routing table অনুযায়ী ঠিক করে প্যাকেটটি কোন output port দিয়ে পাঠাতে হবে।
- এরপর datagram-কে input port → output port-এ সরিয়ে দেয়।

⚠ Note:

Routers শুধুমাত্র IP header দেখে forwarding করে —
তারা TCP/UDP বা application data-এর ভিতরে ঢেকে না।

◊ **Receiver Side**

- ① **Link Layer** ফ্রেম গ্রহণ করে এবং IP datagram উপরের স্তরে পাঠায়।
- ② **Network Layer (IP)** IP header সরিয়ে **transport segment** বের করে।
- ③ **Transport Layer (TCP/UDP)** destination port দেখে ডেটা সঠিক application-এ পাঠায়।

📦 Receiver Flow:

```
Physical Medium → Frame → IP Datagram
      ↓
Network Layer → removes IP header → Segment
      ↓
Transport Layer → removes header → Application Data
```

❖ Network Layer-এর প্রধান কাজ

কাজ

বর্ণনা

Forwarding এক রাউটার থেকে অন্য রাউটারে প্যাকেট সরানো (input → output)।

Routing প্যাকেটের গন্তব্যে পৌঁছানোর সেরা পথ নির্ধারণ করা।

Addressing Source ও Destination IP address বরাদ্দ করা।

Packetizing Transport segment থেকে IP datagram তৈরি করা।

🌐 Network Layer প্রোটোকলসমূহ

প্রোটোকল

IP (Internet Protocol)

কাজ

প্রধান প্রোটোকল, ডেটা প্যাকেটের addressing ও routing পরিচালনা করে।

প্রোটোকল	কাজ
ICMP (Internet Control Message Protocol)	ক্রটি ও কন্ট্রোল বাত্তা পাঠায় (যেমন “destination unreachable”)।
IGMP (Internet Group Management Protocol)	Multicast communication পরিচালনা করে।
ARP (Address Resolution Protocol)	IP address → MAC address ম্যাপ করে (data link layer-এর সহায়তায়)।

□ সারসংক্ষেপ

স্তর	প্রেরকের কাজ	গ্রহণকারীর কাজ
Transport Layer	Segment তৈরি	Segment পুনর্গঠন
Network Layer	Datagram তৈরি ও রাউটিং	Datagram গ্রহণ ও Transport Layer-এ পাঠানো
Routers	Datagram forwarding	(মধ্যবর্তী ধাপ)
Link Layer	Frame প্রেরণ	Frame গ্রহণ

✓ মূল ধারণা

নেটওয়ার্ক লেয়ার **end-to-end delivery** নিশ্চিত করে —
এটি transport layer segment-কে **datagram-এ encapsulate** করে,
router-এর মাধ্যমে পাঠায়,
এবং receiver-এর transport layer-এ পোঁছে দেয়।

🌐 Network Layer-এর দুটি মূল কাজ

❖ ① Forwarding

সংজ্ঞা:

Forwarding হলো একটি রাউটারের ভিতরে প্যাকেটকে ইনপুট লিঙ্ক থেকে আউটপুট লিঙ্কে পাঠানোর প্রক্রিয়া।

❖ অর্থাৎ, একটি প্যাকেট রাউটারের ভিতরে এলে,
রাউটার তার **forwarding table** দেখে ঠিক করে প্যাকেটটি কোন দিক (output port) দিয়ে যাবে।

→ উদাহরণ:

যদি রাউটার দেখে গন্তব্য IP ঠিকানা “192.168.2.5”,
তবে এটি টেবিল থেকে দেখে নেয় যে এটি “Port 3”-এর মাধ্যমে পাঠাতে হবে,
এবং সঙ্গে সঙ্গে প্যাকেটটি ফরওয়ার্ড করে।

⦿ মূল ধারণা:

Forwarding হলো **ডেটা চলাচল** (packet movement) —
“রাউটারের ভিতরের কাজ”।

◦ ২ Routing

সংজ্ঞা:

Routing হলো প্যাকেটটি **কোন পথ** (path) দিয়ে source থেকে destination-এ যাবে তা নির্ধারণের প্রক্রিয়া।

৩ রাউটারগুলো **Routing Algorithm** ব্যবহার করে একে অপরের সাথে তথ্য বিনিময় করে এবং
একটি **routing table** তৈরি করে —
যেখানে গন্তব্য অনুষ্যায়ী সেরা পথ (best route) নির্ধারণ করা থাকে।

→ উদাহরণ:

তুমি যদি ঢাকা থেকে চট্টগ্রাম যাও,
তবে তুমি আগে ঠিক করো **কোন রুটে যাবে** —
ঢাকা → কুমিল্লা → ফেনী → চট্টগ্রাম।
এটাই Routing।

⦿ মূল ধারণা:

Routing হলো **পথ পরিকল্পনা** (path planning) —
“প্যাকেট কোথা দিয়ে যাবে” সেটা নির্ধারণের কাজ।

▣ Analogical Example: Taking a Trip

Network Concept	Real-life Analogy	ব্যাখ্যা
Forwarding	হাইওয়ের একটি ইন্টারচেঞ্জ দিয়ে গাড়ি পার হওয়া	রাউটার এক পোর্ট থেকে অন্য পোর্টে প্যাকেট সরিয়ে দেয়
Routing	ঢাকা থেকে চট্টগ্রাম যাওয়ার পথ ঠিক করা	নেটওয়ার্ক প্যাকেটের গন্তব্যের জন্য সর্বোত্তম পথ নির্ধারণ করে

৩

- Forwarding → Local decision (within one router)
 - Routing → Global decision (across entire network)
-

❖ Routing Algorithms

Routing কাজের জন্য বিভিন্ন algorithm ব্যবহৃত হয়, যেমন —

- Distance Vector Routing (RIP)
- Link State Routing (OSPF)
- Path Vector Routing (BGP)

এগুলো নেটওয়ার্কের অবস্থা অনুযায়ী route নির্ধারণ করে।

□ সারসংক্ষেপ

ফাংশন	স্তর	মূল কাজ	উদাহরণ
Forwarding	রাউটারের ভিতরে	প্যাকেটকে ইনপুট থেকে আউটপুট পোর্টে পাঠানো	রাউটার ট্রাফিক সরানো
Routing	পুরো নেটওয়ার্ক জুড়ে	প্যাকেটের জন্য সর্বেন্ম পথ নির্ধারণ করা	নেটওয়ার্ক মানচিত্র তৈরি

✓ মূল ধারণা

- Forwarding: “এক ধাপ এগিয়ে দেওয়া।”
- Routing: “পুরো যাত্রাপথ ঠিক করা।”

Network Layer এই দুই কাজ একসাথে করে প্যাকেটকে source → routers → destination পর্যন্ত পৌঁছে দেয়।

Network Layer: Data Plane vs Control Plane

নেটওয়ার্ক লেয়ার দুইটি অংশে ভাগ করা যায় —

① Data Plane (Forwarding Plane)

② Control Plane (Routing Plane)

এরা একসাথে কাজ করে প্যাকেটকে source → destination পর্যন্ত পৌঁছে দেয়।

❖ ① Data Plane (ডাটা প্লেন)

□ সংজ্ঞা:

Data Plane হলো রাউটারের ভেতরের লোকাল (local) কাজের অংশ।

এটি নির্ধারণ করে —

“কোন IP datagram কোন input port থেকে আসছে এবং কোন output port দিয়ে যাবে।”

ও মূল কাজ:

- প্যাকেট forwarding করা (input → output port)।
- সিদ্ধান্ত নেওয়া হয় router-এর forwarding table দেখে।
- এটি প্রতিটি প্যাকেটের জন্য ঘটে (per-packet operation)।

❖ বৈশিষ্ট্য:

বৈশিষ্ট্য	বর্ণনা
Scope (ব্যাপ্তি)	Local – এক রাউটারের ভেতরে কাজ করে
কাজের ধরন	প্রতি প্যাকেট অনুযায়ী forwarding
ফলাফল	প্যাকেটটি পরবর্তী রাউটারে পাঠানো হয়
উদাহরণ	“Router: Port 1 থেকে প্যাকেট আসলো → Port 3 দিয়ে পাঠাও।”

❖ সংক্ষেপে:

Data Plane = “প্যাকেট কোথায় যাবে, সেটি বাস্তবায়ন করে।”

❖ ② Control Plane (কন্ট্রোল প্লেন)

□ সংজ্ঞা:

Control Plane হলো **network-wide logic**,
যা নির্ধারণ করে —

“প্যাকেট কোন পথ দিয়ে যাবে (routing path)”
অর্থাৎ, source থেকে destination পর্যন্ত route plan তৈরি করে।

৩ মূল কাজ:

- Routing algorithms চালানো।
- Routing table তৈরি ও আপডেট করা।
- রাউটারগুলোকে একে অপরের সাথে সমন্বয় করতে সাহায্য করা।

ঝঝ বৈশিষ্ট্য:

বৈশিষ্ট্য	বর্ণনা
Scope (ব্যাপ্তি)	Network-wide – পুরো নেটওয়ার্ক জুড়ে কাজ করে
কাজের ধরন	Routing decisions নির্ধারণ
ফলাফল	Forwarding table আপডেট হয়
উদাহরণ	“এই network থেকে এই network-এ যাওয়ার সেরা পথ Route X।”

ক্ষ সংক্ষেপে:

Control Plane = “প্যাকেট কোথায় যাবে, সেটি পরিকল্পনা করে।”

ক্ষ দুই Control Plane Approach

① Traditional Routing Algorithms

- প্রতিটি রাউটার নিজের মধ্যে routing algorithm (RIP, OSPF, BGP) চালায়।
- তারা একে অপরের সাথে তথ্য বিনিময় করে।
- প্রতিটি রাউটারের নিজস্ব routing table থাকে।

→ Implemented inside routers.

২ Software-Defined Networking (SDN)

- Routing logic আলাদা (remote) server-এ থাকে।
- Server (SDN controller) পুরো নেটওয়ার্কের routing সিদ্ধান্ত নেয়।
- Routers কেবল forwarding-এর কাজ করে।

→ Implemented in centralized servers.

□ Data Plane vs Control Plane: তুলনামূলক টেবিল

বিষয়	Data Plane	Control Plane
কাজের ধরন	Packet forwarding	Path determination
Scope	Local (within router)	Network-wide
Function	Moves packets	Plans routes
Algorithm	Forwarding lookup	Routing algorithm
উদাহরণ	IP forwarding	RIP, OSPF, BGP
বাস্তবায়ন	Routers-এর ভিতরে	Routers বা SDN controllers-এ

ঔষধ উদাহরণ (Analogy: Taking a Trip)

Network Concept	বাস্তব উদাহরণ
Control Plane	মানচিত্র দেখে পুরো রুট পরিকল্পনা করা (ঢাকা → কুমিল্লা → ফেনী → চট্টগ্রাম)
Data Plane	প্রতিটি রাস্তার মোড়ে সিদ্ধান্ত নেওয়া – “এখানে বামে যাবো না ডানে?”

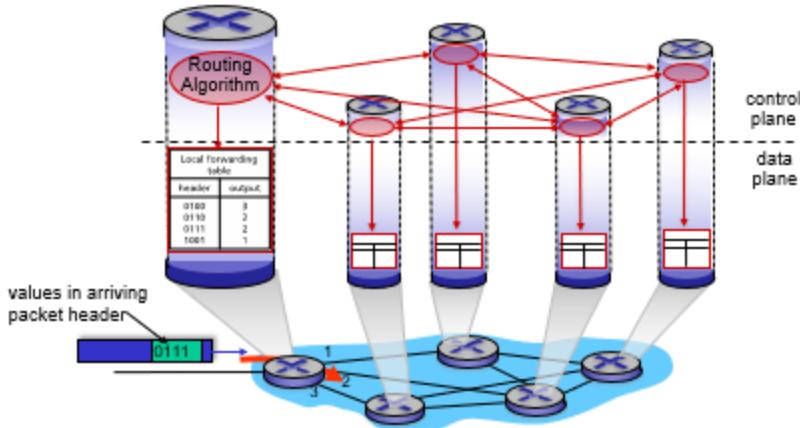
✓ মূল ধারণা

- Control Plane: রুট পরিকল্পনা করে।
- Data Plane: প্যাকেটকে সেই রুটে চলতে সাহায্য করে।

অর্থাৎ, Control Plane “brains” ☑ আর Data Plane “hands” ⚡

Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane



⦿ Per-Router Control Plane কী

Per-Router Control Plane মানে হলো —

প্রতিটি রাউটার নিজের ভেতরে **routing algorithm** চালায় এবং নিজস্ব **routing table** তৈরি করে।

→ অর্থাৎ, প্রতিটি রাউটার নিজে “**বুদ্ধিমান (intelligent)**”,
সে জানে কোন গন্তব্যে প্যাকেট পাঠাতে হবে এবং কোন পোর্ট দিয়ে ফরওয়ার্ড করতে হবে।

❖ Control Plane এবং Data Plane-এর সম্পর্ক

চিত্রে তুমি দেখতে পাচ্ছ যে প্রতিটি রাউটার দুটি ভাগে বিভক্ত —

① **Control Plane (উপরের অংশ)**

② **Data Plane (নিচের অংশ)**

Plane	কাজ
Control Plane	রাউটার কীভাবে forwarding সিদ্ধান্ত নেবে তা নির্ধারণ করে (routing algorithm চালায়)
Data Plane	ঐ সিদ্ধান্ত অনুযায়ী প্যাকেট আসা-যাওয়া (forwarding) করে

ঔক্ষণ্য Control Plane-এর কাজ (উপরের অংশ)

- প্রতিটি রাউটারের মধ্যে routing algorithm component থাকে।
- রাউটারগুলো একে অপরের সাথে routing information exchange করে।
- এই তথ্য থেকে প্রতিটি রাউটার নিজের routing table তৈরি করে।
- Routing table control plane-এর ফলাফল,
যা data plane-এ ব্যবহৃত হয় forwarding-এর জন্য।

ঔক্ষণ্য Routing Algorithms:

- Distance Vector (যেমন RIP)
- Link State (যেমন OSPF)
- Path Vector (যেমন BGP)

ঔক্ষণ্য Data Plane-এর কাজ (নিচের অংশ)

- এটি রিলেল-টাইমে প্যাকেট গ্রহণ করে এবং forwarding table দেখে সঠিক আউটপুট পোর্ট নির্ধারণ করে।
- উদাহরণস্বরূপ:
- Destination IP: 10.1.1.5 → Forward to Port 2
- Data Plane এই forwarding টেবিলটি Control Plane থেকে পায়।

ঔক্ষণ্য Data Plane = "Action"

ঔক্ষণ্য Control Plane = "Decision"

চিত্র বিশ্লেষণ (Figure Explanation)

চিত্রে দেখা যাচ্ছে:

- প্রতিটি রাউটারের মধ্যে একটি Routing Algorithm রয়েছে (উপরের অংশে)।
- রাউটারগুলো লাল লাইন দিয়ে পরস্পরের সাথে সংযুক্ত — এটি Control Plane Communication নির্দেশ করছে।
- নিচে নীল অংশে Data Plane কাজ করছে, যেখানে প্যাকেট আসছে (values in arriving packet header)।
- Control Plane উপরে থেকে Data Plane-এ নির্দেশ দেয় কোন প্যাকেট কোথায় ফরওয়ার্ড হবে।

☒ Per-Router Control Plane-এর সারসংক্ষেপ

দিক	বর্ণনা
Location	প্রতিটি রাউটারের ভেতরে
Function	Routing algorithm চালায় ও forwarding table তৈরি করে
Coordination	অন্য রাউটারগুলোর সাথে routing তথ্য বিনিময় করে
Output	Forwarding table (যা data plane ব্যবহার করে)
Benefit	প্রতিটি রাউটার স্বতন্ত্রভাবে কাজ করতে পারে
Example Protocols	RIP, OSPF, BGP

☒ সহজভাবে উদাহরণ

ধরা যাক, শহরের প্রতিটি ট্রাফিক সিগন্যালের মধ্যে একটি ছোট কম্পিউটার আছে যা নিজের মতো করে সিদ্ধান্ত নিচ্ছে:
“যদি এই রাস্তায় ভিড় বেশি, তাহলে অন্য রাস্তায় গাড়ি পাঠাও।”

এটাই Per-Router Control Plane —
প্রতিটি রাউটার নিজেই নিজের “routing intelligence” বহন করে।

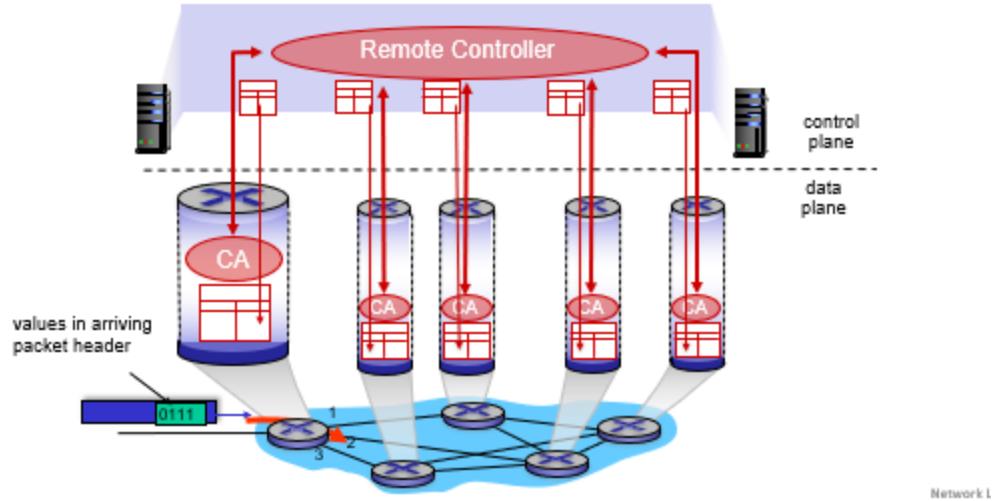
☒ মূল ধারণা

Per-Router Control Plane:

প্রতিটি রাউটার নিজস্বভাবে routing algorithm চালায়,
নিজের forwarding table তৈরি করে,
এবং Data Plane-কে নির্দেশ দেয় কীভাবে প্যাকেট ফরওয়ার্ড করতে হবে।

Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



🌐 Software-Defined Networking (SDN) Control Plane

❖ মূল ধারণা

SDN (Software-Defined Networking) হলো এমন এক নেটওয়ার্ক স্থাপনা যেখানে **control plane** (routing decision) আলাদা করে একটি **centralized controller**-এ রাখা হয় এবং **routers/switches** শুধুমাত্র **data plane** (forwarding) হিসেবে কাজ করে।

৩. অর্থাৎ:

“বুদ্ধি (intelligence)” কেন্দ্রীভূত হয় Controller-এ,
আর “কাজ (forwarding)” করে রাউটার/সুইচগুলো।

ঔঙ্গ চিত্র ব্যাখ্যা (Figure Explanation)

চিত্রে দেখা যাচ্ছে:

- উপরে একটি **Remote Controller** আছে — এটি হলো SDN-এর “মাস্তিষ্ক” ঔঙ্গ।
- নিচে আছে একাধিক **router/switch**, প্রতিটির মধ্যে **CA (Control Agent)** নামে ছোট সফটওয়্যার কম্পোনেন্ট আছে।

- নিচের নীল অংশ হলো **Data Plane**, যেখানে প্যাকেট আসা-যাওয়া করছে।
 - Controller উপরে থেকে প্রত্যেকটি রাউটার/সুইচে **forwarding rules / tables** পাঠাচ্ছে (লোল তীর দ্বারা দেখানো)।
 - প্রতিটি রাউটার সেই forwarding rule অনুসরণ করে প্যাকেট ফরওয়ার্ড করছে।
-

❖ SDN Control Plane-এর কাজের ধাপসমূহ

① Network Information Gathering:

প্রতিটি রাউটার তার অবস্থা (status, traffic, topology info) central controller-এ পাঠায়।

② Decision Making (Brain Work):

Controller সেই তথ্য ব্যবহার করে routing decision ও forwarding table তৈরি করে।

③ Instruction Distribution:

Controller প্রত্যেক রাউটার বা সুইচে forwarding rules পাঠিয়ে দেয়।

④ Packet Forwarding:

এখন রাউটারগুলো সেই forwarding rules অনুযায়ী data plane-এ প্যাকেট পাঠায়।

🔍 Traditional বনাম SDN Control Plane তুলনা

বৈশিষ্ট্য	Traditional (Per-Router)	SDN (Centralized)
Control Location	প্রতিটি রাউটারের ভিতরে	এক কেন্দ্রীয় controller-এ
Routing Decision	স্থানীয়ভাবে (local) নেওয়া	গ্লোবালভাবে (network-wide) নেওয়া
Forwarding Table তৈরি করে কে?	রাউটার নিজেই	Controller তৈরি করে
Scalability	সীমিত (distributed decisions)	বেশি (centralized management)
Management	জটিল (প্রত্যেক রাউটার আলাদা সহজ (controller থেকে এক জায়গা কনফিগার করতে হয়)	থেকে সব নিয়ন্ত্রণ করা যায়)
Example Protocol	RIP, OSPF, BGP	OpenFlow (SDN protocol)

□ OpenFlow — SDN Communication Protocol

SDN controller এবং routers/switches-এর মধ্যে যোগাযোগের জন্য

OpenFlow protocol ব্যবহৃত হয়।

এটি নির্দেশ দেয় —

কোন প্যাকেট কোন header অনুযায়ী কোন port দিয়ে যাবে।

ক্ষেত্র CA (Control Agent) কী?

- প্রতিটি router/switch-এর মধ্যে ছোট একটি সফটওয়্যার মডিউল থাকে, যাকে বলে **Control Agent (CA)**।
- এটি controller-এর নির্দেশ গ্রহণ করে এবং তা router-এর forwarding hardware-এ প্রয়োগ করে।

⇒ Controller → (Rules পাঠায়) → CA → (Applies to) → Data Plane

□ SDN-এর সুবিধাসমূহ

সুবিধা

Centralized intelligence এক জায়গা থেকে পুরো network নিয়ন্ত্রণ করা যায়।

Programmability Network কে software দিয়ে সহজে কনফিগার করা যায়।

Flexibility Traffic flow ডাইনামিকভাবে পরিবর্তন করা যায়।

Simplified Management আলাদা আলাদা router configure করতে হয় না।

Faster innovation নতুন network services সহজে implement করা যায়।

বর্ণনা

ঔষধ উদাহরণ (Analogy)

ধরা যাক তুমি একটি শহরের ট্রাফিক কন্ট্রোল সিস্টেম ডিজাইন করছো:

- Traditional System:**
প্রতিটি ট্রাফিক সিগন্যাল নিজেই সিদ্ধান্ত নেয় কখন লাল/সবুজ দেখাবে।
(অর্থাৎ, per-router control plane)
 - SDN System:**
একটি কেন্দ্রীয় ট্রাফিক অফিস (controller) আছে,
যা পুরো শহরের সিগন্যালগুলোকে একসাথে নিয়ন্ত্রণ করছে।
(অর্থাৎ, SDN control plane)
-

মূল ধারণা

Software-Defined Networking (SDN)-এ
Control Plane কেন্দ্রীভূতভাবে Remote Controller-এ থাকে,
এবং Data Plane শুধু forwarding-এর কাজ করে।

Controller forwarding tables গণনা করে
এবং routers-এ ইনস্টল করে দেয়।

Network Service Model কী?

◦ সংজ্ঞা:

Network Service Model হলো এমন একটি ধারণা যা নির্ধারণ করে
নেটওয়ার্ক লেয়ার কীভাবে datagram (packet) গুলোকে প্রেরক (sender) থেকে গ্রাহক (receiver)-
এর কাছে পৌঁছে দেবে —
অর্থাৎ, এটি বর্ণনা করে channel বা communication path-এর quality এবং behavior।

৩ সহজভাবে:

“Sender থেকে Receiver-এ ডেটা যাওয়ার পথে নেটওয়ার্ক কী ধরনের নিশ্চয়তা বা সুবিধা দিচ্ছে”
— সেটিই Network Service Model।

শুটি দুটি স্তরে পরিষেবা প্রদান করা যায়

① For Individual Datagrams

(একটি প্যাকেট এককভাবে কীভাবে ডেলিভারি হবে)

② For a Flow of Datagrams

(একটি ধারাবাহিক প্যাকেট প্রবাহ বা data stream কীভাবে পরিচালিত হবে)

⌚ Example Services for Individual Datagrams

পরিষেবা	বর্ণনা
Guaranteed Delivery	প্রতিটি প্যাকেট নিশ্চয়তার সাথে গন্তব্যে পৌঁছাবে (lost হবে না)।
Guaranteed Delivery with less than 40 msec delay	প্যাকেট গন্তব্যে সর্বাধিক ৪০ মিলিসেকেন্ডের মধ্যে পৌঁছাবে (low-latency service)।
ঢ় এই ধরণের পরিষেবা real-time applications (যেমন VoIP, online gaming)-এর জন্য গুরুত্বপূর্ণ।	

⌚ Example Services for a Flow of Datagrams

পরিষেবা	বর্ণনা
In-order Datagram Delivery	প্যাকেটগুলো ঠিক সেই ক্রমে পৌঁছাবে যেভাবে পাঠানো হয়েছিল।
Guaranteed Minimum Bandwidth to Flow	নির্দিষ্ট data rate (যেমন 2 Mbps) নিশ্চিত করা হবে ওই সংযোগের জন্য।
Restrictions on Changes in Inter-Packet Spacing	প্যাকেটগুলোর মধ্যে সময় ব্যবধান (timing) একই রকম থাকবে — বড় পরিবর্তন হবে না (useful for audio/video streaming)।

ঢ় এই ধরণের পরিষেবা **streaming** ও **multimedia communication**-এর জন্য ব্যবহৃত হয়।

⌚ Simplified View

ধরন	পরিষেবার উদাহরণ	ব্যবহারের ক্ষেত্র
Individual Datagram	Guaranteed delivery, low delay	Short transactions (যেমন DNS, Ping)
Flow of Datagrams	In-order, fixed spacing, guaranteed rate	Streaming, VoIP, Video conferencing

⌚ বাস্তব নেটওয়ার্কে কী ঘটে?

ইন্টারনেটের IP নেটওয়ার্কে, এই পরিষেবাগুলোর বেশিরভাগই **প্রদত্ত** নয়।

IP একটি **best-effort delivery service** ব্যবহার করে —

অর্থাৎ:

“প্যাকেট পাঠাবে হবে, কিন্তু পৌঁছাবে কি না, কখন পৌঁছাবে, বা কোন ক্রমে পৌঁছাবে — তার নিশ্চয়তা নেই।”

তবে কিছু বিশেষ নেটওয়ার্ক (যেমন MPLS, ATM, বা 5G QoS) এই **service guarantees** প্রদান করে।

শুরু মূল ধারণা (Key Takeaway)

Network Service Model নির্ধারণ করে নেটওয়ার্ক কর্তৃতা “বিশ্বাসযোগ্য” এবং “দ্রুত” ডেটা পাঠাবে।

- **Individual Datagram Services:** প্রতিটি প্যাকেটের নিশ্চয়তা।
- **Flow-based Services:** ধারাবাহিক ডেটা প্রবাহের গুণমান ও timing নিয়ন্ত্রণ।

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet “best effort” service model

No guarantees on:

- i. successful datagram delivery to destination
- ii. timing or order of delivery
- iii. bandwidth available to end-end flow

🌐 Network-Layer Service Model

নেটওয়ার্ক লেয়ার নির্ধারণ করে ডেটাগ্রামগুলো কীভাবে sender থেকে receiver-এ পরিবাহিত হবে।

এই পরিষেবার ধরন বা আচরণকে বলা হয় **Service Model**।

⌚ Internet-এর Service Model: “Best-Effort”

◊ সংজ্ঞা:

Best-Effort Service Model অর্থাৎ,
ইন্টারনেট চেষ্টা করবে প্যাকেট পৌঁছে দিতে,
কিন্তু কোনো নিশ্চয়তা দেবে না যে তা সফলভাবে, নির্দিষ্ট সময়ে বা সঠিক ক্রমে পৌঁছাবে।

ঔ সহজভাবে:

“We will try our best to deliver your packet — but we can’t promise anything.”

❖ Quality of Service (QoS) Guarantees

বৈশিষ্ট্য	গ্যারান্টি আছে?	ব্যাখ্যা
Bandwidth (প্রস্তুতি)	✗ None	নির্দিষ্ট bandwidth নিশ্চিত করা হয় না।
Loss (ক্ষতি)	✗ No	কিছু প্যাকেট হারিয়ে যেতে পারে।
Order (ক্রম)	✗ No	প্যাকেটগুলো পাঠানোর ক্রমে পৌঁছাবে তার নিশ্চয়তা নেই।
Timing (সময়)	✗ No	বিলম্ব (delay) পরিবর্তন হতে পারে; কোনো সময়সীমা নেই।

□ “Best-Effort” Model-এ কোনো নিশ্চয়তা নেই

Internet “Best-Effort” service model-এ নিচের বিষয়ে কোনো গ্যারান্টি নেই ঙু

① Successful Delivery

- প্যাকেট গন্তব্যে পৌঁছাবে — তার নিশ্চয়তা নেই।

② Timing বা Order of Delivery

- প্যাকেট সঠিক ক্রমে বা নির্দিষ্ট সময়ে পৌঁছাবে — তার নিশ্চয়তা নেই।

৩ Bandwidth Availability

- পুরো সংযোগে (end-to-end) নির্দিষ্ট ডেটা রেট নিশ্চিত নয়।
-

ক্ষেত্র উদাহরণে বোঝা

ধরা যাক তুমি Gmail-এ ইমেইল পাঠালে বা YouTube-এ ভিডিও দেখলে —
ইন্টারনেট তার সর্বেচ চেষ্টা করে প্যাকেট পাঠাতে,
কিন্তু যদি কিছু প্যাকেট হারিয়ে যায় বা দেরি হয়,
তাহলে সেই সমস্যা অ্যাপ্লিকেশন লেয়ারে (যেমন TCP, YouTube buffer) সমাধান করতে হয়।

→ অর্থাৎ,
IP layer শুধুমাত্র “delivery attempt” করে,
“reliability guarantee” দেয় না।

১. কেন এটাকে “Best-Effort” বলা হয়?

কারণ নেটওয়ার্ক:

- প্যাকেট drop হতে পারে, delay হতে পারে।
 - কিন্তু যতদূর সম্ভব এগুলো minimize করার চেষ্টা করে।
 - যদি congestion বা link failure ঘটে, তখন packet discard করা হয়।
-

২ Comparison with Reliable Models

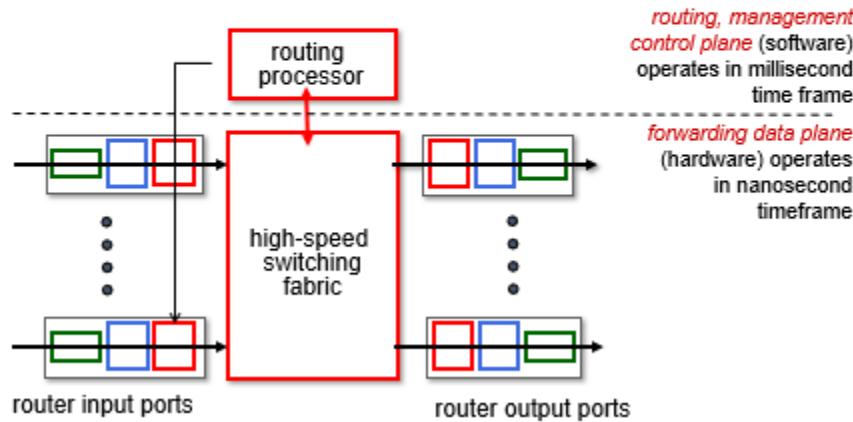
মডেল	ডেলিভারি নিশ্চয়তা	টাইমিং নিশ্চয়তা	ব্যান্ডউইথ নিশ্চয়তা	উদাহরণ
Best-Effort (IP)	✗ না	✗ না	✗ না	Internet
Guaranteed Service (QoS)	✓ হ্যাঁ	✓ হ্যাঁ	✓ হ্যাঁ	ATM, MPLS, 5G networks

৩ মূল ধারণা

Internet-এর Network Layer (IP) হলো **Best-Effort Service Model**,
যেখানে কোনো QoS guarantee নেই —
কিন্তু এটি সাধারণ, সহজ, এবং স্কেলেবল বলে পুরো ইন্টারনেট এভাবেই কাজ করে।

Router architecture overview

high-level view of generic router architecture:



🌐 Router Architecture Overview (রাউটার স্থাপত্যের সারাংশ)

রাউটার একটি বিশেষ নেটওয়ার্ক ডিভাইস,
যার প্রধান কাজ হলো প্যাকেটকে **input port** → **output port**-এ স্থানান্তর করা
এবং সঠিক পথে (route) পাঠানো।

রাউটারকে সাধারণভাবে দুটি প্রধান অংশে ভাগ করা হয় —

- 1 Control Plane
- 2 Data Plane

❖ High-Level Components

চিত্রে চারটি গুরুত্বপূর্ণ অংশ দেখা যাচ্ছে ছবি

-
- 1 Router Input Ports
 - 2 Router Output Ports
 - 3 High-Speed Switching Fabric
 - 4 Routing Processor
-

❖ 1 Input Ports

- এখানে প্যাকেটগুলো প্রথমে আসে (যখন তারা রাউটারে প্রবেশ করে)।
- ইনপুট পোর্টগুলো incoming link থেকে প্যাকেট গ্রহণ করে এবং সেগুলোর header পরীক্ষা করে forwarding decision এর জন্য প্রস্তুত করে।
- Forwarding table অনুসারে packet-কে কোন output port-এ পাঠাতে হবে তা নির্ধারণ করে।

⌚ কাজের গতি: Nanosecond সময়সীমায় — খুব দ্রুত।

❖ 2 Switching Fabric (মধ্যবর্তী সংযোগ)

- এটি হলো রাউটারের “heart” ❤ — যেখানে প্যাকেট ইনপুট পোর্ট থেকে আউটপুট পোর্টে পাঠানো হয়।
- এটি High-Speed Hardware দ্বারা গঠিত।
- প্যাকেটের অভ্যন্তরীণ রুট নির্ধারণ করে যাতে কোনো congestion না হয়।

💡 উদাহরণ: Crossbar switch, Shared bus, বা Interconnection network।

❖ 3 Output Ports

- Switching fabric থেকে আসা প্যাকেটগুলোকে আউটগোয়িং লিঙ্কে পাঠায়।
- এখানে প্যাকেট কিউ (queue) হতে পারে যদি আউটপুট লিঙ্ক ব্যস্ত থাকে।
- Packet scheduling ও buffer management এখানে ঘটে।

⌚ কাজের গতি: Nanosecond সময়সীমায়।

❖ 4 Routing Processor

- এটি হলো রাউটারের মস্তিষ্ক (brain) ।
- Control Plane-এ কাজ করে।
- Routing algorithms চালায় (যেমন RIP, OSPF, BGP) এবং **routing table** তৈরি করে।
- সেই routing table-এর ভিত্তিতে **forwarding table** তৈরি করে Data Plane-এ পাঠায়।

□ **কাজের সময়:** Millisecond সময়সীমায় — কারণ এটি software-ভিত্তিক অপারেশন।

🧠 Control Plane বনাম Data Plane

দিক	Control Plane	Data Plane
মূল কাজ	Routing decision, management	Actual packet forwarding
অবস্থান	Routing Processor	Input/Output Ports & Switching Fabric
বাস্তবায়ন	Software	Hardware
সময়সীমা	Milliseconds	Nanoseconds
উদাহরণ	Routing algorithms (OSPF, BGP)	High-speed packet switching

□ Packet Flow Summary

```
Incoming Packet
  ↓
Router Input Port (header check)
  ↓
Switching Fabric (moves packet inside router)
  ↓
Router Output Port (queues + sends)
  ↓
Outgoing Link → Next Router or Destination
```

🔍 Key Takeaway

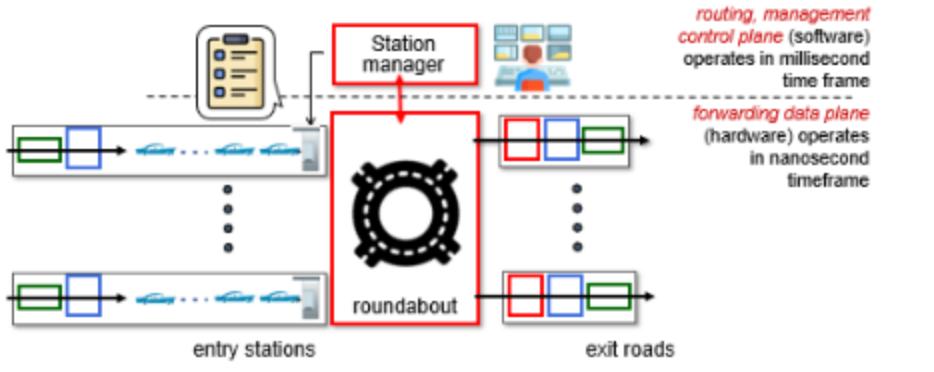
একটি রাউটার দুটি প্রধান কাজ করে —

- **Control Plane:** সিদ্ধান্ত নেয় (software-based, ধীর কিন্তু বুদ্ধিমান)
- **Data Plane:** forwarding করে (hardware-based, দ্রুত ও কার্যকর)

Routing Processor “মস্তিষ্ক” । আর Switching Fabric ও Ports “হাত-পা” ।

Router architecture overview

analogy view of generic router architecture:



Router Architecture Analogy: The Roundabout Example

এই ছবিতে রাউটারের কাজকে একটি রাস্তার রাউন্ডঅ্যাবাউট (roundabout)-এর সাথে তুলনা করা হয়েছে।

এটি রাউটারের ভেতরের **control plane** এবং **data plane** ধারণাকে সহজভাবে ব্যাখ্যা করে।

Analogy-এর উপাদানগুলো

রাউটার অংশ	উদাহরণে যা দ্বারা বোঝানো হয়েছে	কাজের ব্যাখ্যা
Entry Stations (Router Input Ports)	প্রবেশ রাস্তা	এখানে গাড়িগুলো (প্যাকেট) রাউন্ডঅ্যাবাউটে ঢোকে
Roundabout (Switching Fabric)	রাউটারের ভিতরের সংযোগ ব্যবস্থা	গাড়িগুলো (প্যাকেট) ঘুরে সঠিক পথে (output) যায়
Exit Roads (Router Output Ports)	বের হওয়ার রাস্তা	যেখানে গাড়িগুলো (প্যাকেট) গন্তব্যের দিকে রওনা হয়
Station Manager (Routing Processor)	ট্রাফিক কন্ট্রোলার বা ম্যানেজার	সিদ্ধান্ত নেয় কোন গাড়ি কোন রাস্তা দিয়ে যাবে

Router-এর দুইটি প্রধান স্তর

❖ 1 Control Plane (Software Level)

- এটি হলো রাউটারের “বুদ্ধি” ক্ষেত্র।
- সিদ্ধান্ত নেয়: “এই প্যাকেট কোন পথে যাবে?”
- এটি software দ্বারা পরিচালিত হয় — **routing algorithms** (যেমন RIP, OSPF, BGP) ব্যবহার করে।
- সময় লাগে milliseconds — কারণ এটি চিন্তা ও পরিকল্পনার কাজ।

ও Analogies:

Station Manager বা ট্রাফিক পুলিশ যেমন সিদ্ধান্ত নেয় কোন গাড়ি কোন রাস্তা পাবে।

❖ 2 Data Plane (Hardware Level)

- এটি হলো “অ্যাকশন” বা বাস্তব forwarding অংশ।
- Control plane থেকে পাওয়া নির্দেশ অনুসারে প্যাকেটকে দ্রুত সঠিক আউটপুটে পাঠায়।
- এটি hardware দ্বারা পরিচালিত হয় এবং nanosecond সময়ের মধ্যে কাজ করে।

ও Analogies:

রাউন্ডঅ্যাবাউটে গাড়িগুলো নিজের লেনে ঘুরে সঠিক রাস্তায় বেরিয়ে যাচ্ছে — কোনো থামানো বা চিন্তা ছাড়াই।

□ Router-এর কাজের ধাপ (In Analogy Terms)

1 গাড়ি (packet) entry road (input port) দিয়ে রাউন্ডঅ্যাবাউটে ঢোকে।

2 Station manager (control plane) দেখে নেয় গাড়ি কোথায় যাবে।

3 গাড়ি roundabout (switching fabric) ঘুরে সঠিক exit road (output port) এ যায়।

4 Exit road দিয়ে গাড়ি তার পরবর্তী গন্তব্যে চলে যায়।

⌚ Timing Difference

Plane	কাজের ধরন	সময়কাল
Control Plane	Routing & Management (Software)	Milliseconds
Data Plane	Forwarding (Hardware)	Nanoseconds

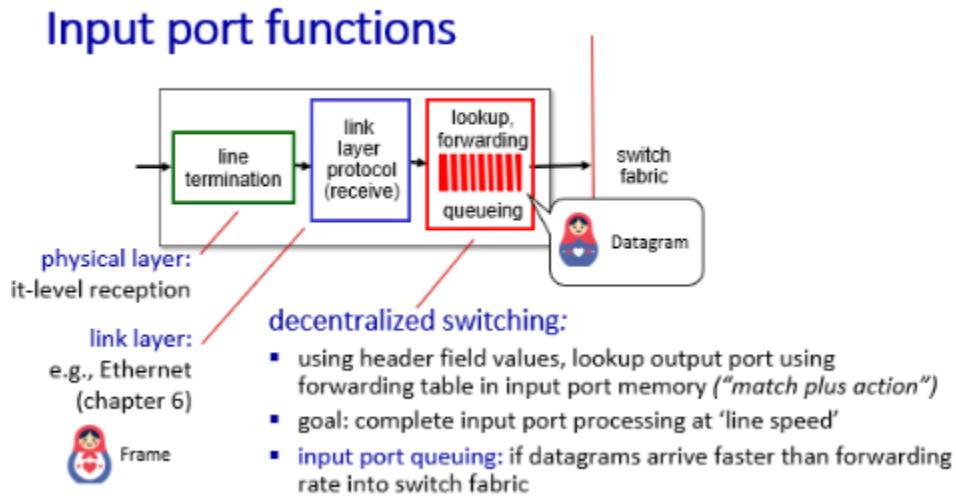
অর্থাৎ —

Control plane চিন্তা করে, Data plane কাজ করে।

✓ মূল ধারণা

রাউটারকে একটি ব্যস্ত **রাউন্ডঅ্যাবাউট** হিসেবে ভাবো —
যেখানে গাড়িগুলো (প্যাকেট) দ্রুত আসে-যায়,
কিন্তু তাদের দিকনির্দেশনা দেয় একটি **স্টেশন ম্যানেজার** (routing processor)।

Control Plane সিদ্ধান্ত নেয় 
Data Plane তা বাস্তবায়ন করে 



🌐 Input Port Functions (ইনপুট পোর্টের কাজ)

রাউটারের **input port** হলো সেই জায়গা যেখানে প্যাকেট প্রথম প্রবেশ করে।
এখানেই শুরু হয় রাউটারের ভিতরে প্যাকেট প্রক্রিয়াকরণ।

❖ Input Port-এর প্রধান ধাপসমূহ

চির অনুযায়ী, একটি ইনপুট পোর্ট চারটি মূল ধাপে কাজ করে ছে

1 Line Termination (Physical Layer)

- এই ধাপে প্যাকেটের বিট লেভেল রিসেপশন (bit-level reception) হয়।
- অর্থাৎ, নেটওয়ার্কের ফিজিক্যাল মিডিয়াম (যেমন Ethernet cable, fiber optic) থেকে আসা সিগন্যালগুলোকে ডিজিটাল ডেটায় রূপান্তর করা হয়।
- এটি Physical Layer (Layer 1)-এর কাজ।

৩ উদাহরণ:

Ethernet PHY chip সিগন্যাল থেকে বিট বের করে নেয়।

2 Link-Layer Protocol (Data Link Layer)

- এরপর আসে Link Layer Processing (যেমন Ethernet protocol)।
- এটি ফ্রেম (Frame) গ্রহণ করে এবং ভেতরের প্যাকেট বের করে আনে।
- এখানে MAC address, CRC ইত্যাদি ঘাচাই হয়।
- এটি Data Link Layer (Layer 2)-এর কাজ।

৩ উদাহরণ:

Ethernet ফ্রেম থেকে IP datagram বের করা।

3 Lookup, Forwarding, and Queuing (Network Layer)

এটি সবচেয়ে গুরুত্বপূর্ণ অংশ — Network Layer-এর কাজ।

❖ *Lookup:*

- প্যাকেটের header (যেমন destination IP address) দেখে রাউটার Forwarding Table থেকে খুঁজে বের করে কোন output port দিয়ে পাঠাতে হবে।
- এই কাজটিকে বলে Decentralized Switching, কারণ প্রতিটি input port নিজে এই lookup করে নেয় (central processor-এর সাহায্য ছাড়াই)।
- পদ্ধতিটি হলো “Match + Action”:
 - Match: Header field মিলিয়ে দেখা।
 - Action: কোন পোর্টে পাঠাতে হবে তা নির্ধারণ।

◊ *Forwarding:*

- একবার lookup সম্পন্ন হলে, প্যাকেটটি switching fabric-এ পাঠানো হয় যাতে সেটি নির্ধারিত output port-এ যেতে পারে।

◊ *Queuing:*

- যদি input rate (যে গতিতে প্যাকেট আসছে) switch fabric-এর forwarding rate-এর চেয়ে বেশি হয়, তখন প্যাকেটগুলো input queue-তে জমা থাকে।
- এই কিউ (queue) congestion নিয়ন্ত্রণে সাহায্য করে।

৩ উদাহরণ:

যদি প্যাকেট অনেক দ্রুত আসে কিন্তু switch fabric থেকে পাঠায়, তাহলে প্যাকেট queue-তে দাঁড়িয়ে অপেক্ষা করে।

4 Switch Fabric Interface

- এটি input port এবং switching fabric-এর সংযোগস্থল।
 - এখান দিয়ে প্যাকেট fabric-এর ভিতরে প্রবেশ করে এবং তারপর সঠিক output port-এ পাঠানো হয়।
-

-Decentralized Switching-এর গুরুত্ব

বৈশিষ্ট্য

বর্ণনা

কাজের ধরন প্রতিটি input port নিজে forwarding lookup করে

উদ্দেশ্য উচ্চ গতি (line speed) বজায় রাখা

সুবিধা রাউটারকে parallelভাবে প্যাকেট প্রসেস করতে দেয়

ফলাফল দ্রুত forwarding, কম delay

↳ Timing Perspective

স্তর	কাজ	সময়সীমা
Physical Layer	Bits গ্রহণ	Nanoseconds
Link Layer	ফ্রেম থেকে প্যাকেট আলাদা	Nanoseconds
Network Layer (Lookup)	Output port নির্ধারণ	Nanoseconds
Control Plane (Routing Decision)	Routing table আপডেট	Milliseconds

☒ সংক্ষেপে প্যাকেট প্রবাহ

Frame (Physical medium)

↓

Line Termination → Link Layer Protocol → Lookup/Forwarding → Queue
 ↓
 Switch Fabric → Output Port → Next Hop

☑ মূল ধারণা

রাউটারের ইনপুট পোর্ট প্যাকেটকে গ্রহণ করে,
 ডিকোড করে, output port নির্ধারণ করে,
 এবং switch fabric-এ পাঠায় —
 সবকিছু line speed-এ (hardware-level parallel processing) সম্পন্ন হয়।

Destination-based forwarding

forwarding table	
Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through	n
11001000 00010111 00010000 00000100 through	3
11001000 00010111 00010000 00000111	
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

🌐 Destination-Based Forwarding কী?

❖ সংজ্ঞা:

Destination-based forwarding হলো একটি পদ্ধতি যেখানে রাউটার
প্যাকেটের গন্তব্য (destination IP address) দেখে ঠিক করে
কোন output link/interface দিয়ে প্যাকেটটি পাঠানো হবে।

❖ সহজভাবে:

“এই IP address কোন দিকে যাবে?” —
সেটি forwarding table দেখে নির্ধারণ করা হয়।

❖ Forwarding Table (রাউটারের রুট ম্যাপ)

চিত্রে বাম পাশে আছে Destination Address Range,
আর ডান পাশে আছে Link Interface (যেখানে প্যাকেট পাঠানো হবে)।

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 → 11001000 00010111 00010000 00000100	0
11001000 00010111 00010000 00000101 → 11001000 00010111 00010000 00000111	3
11001000 00010111 00010000 00001000 → 11001000 00010111 00010000 11111111	3
11001000 00010111 00011001 00000000 → 11001000 00010111 00011111 11111111	2
Otherwise (যদি উপরের কোন রেঞ্জে না পড়ে)	3

🔍 Forwarding-এর কাজের ধাপ

- 1 প্যাকেট রাউটারে আসে।
- 2 রাউটার destination IP address দেখে।
- 3 রাউটার forwarding table-এ এই address মেলানোর চেষ্টা করে।
- 4 যদি পাওয়া গেলে, প্যাকেট নির্দিষ্ট output link/interface-এ পাঠানো হয়।
- 5 যদি কোনো মিল না পাওয়া যায়, তাহলে default route (“otherwise”) ব্যবহার করা হয়।

⦿ Example বোঝার জন্য:

ধরা যাক, কোনো প্যাকেটের destination IP address হলো

11001000 00010111 00010000 00000010

রাউটার টেবিলে দেখে এটি কোন রেঞ্জে পড়ে ছে

→ এটি প্রথম রেঞ্জের মধ্যে পড়ে

(11001000 00010111 00010000 00000000 → 11001000 00010111 00010000 00000100)

তাহলে প্যাকেটটি যাবে **Link Interface 0** দিয়ে।

⦿ Key Point: Range-based Matching

- টেবিলে প্রতিটি রেঞ্জ একটি **network prefix** বা **subnet** নির্দেশ করে।
 - রাউটার এই prefix-এর সাথে IP address-এর প্রথম কয়েকটি বিট মেলায়।
 - রেঞ্জ যদি সুন্দরভাবে বিভক্ত না হয় (যেমন 8-বিটের ব্লক নয়),
তাহলে **Longest Prefix Matching (LPM)** ব্যবহার করা হয়।
-

⦿ Longest Prefix Matching (LPM) কী?

স্লাইডের নিচের প্রশ্নে বলা আছে ছে

“But what happens if ranges don’t divide up so nicely?”

এটি বোঝায় —

সব নেটওয়ার্ক রেঞ্জ সমান আকারে ভাগ করা যায় না।

সেক্ষেত্রে রাউটার **Longest Prefix Match** করে, অর্থাৎ:

যে prefix (নেটওয়ার্ক অংশ) প্যাকেটের IP address-এর সাথে সবচেয়ে বেশি বিটে মিলে,
সেই রুটটি বেছে নেওয়া হয়।

৩ উদাহরণ:

11001000 00010111 00010000 00000100

এই IP যদি দুটি subnet-এ মিলে যায়, তাহলে
রাউটার সেই subnet বেছে নেবে যার prefix বেশি দীর্ঘ (অর্থাৎ আরও নির্দিষ্ট)।

□ Destination-Based Forwarding-এর সারাংশ

ধাপ	কাজ
1	প্যাকেট আসে input port-এ
2	Destination IP address পরীক্ষা
3	Forwarding table থেকে match খোঁজা
4	মিলে গেলে নির্দিষ্ট output link নির্ধারণ
5	না মিললে default route ব্যবহার

মূল ধারণা

Destination-based forwarding-এ রাউটার কেবলমাত্র প্যাকেটের গন্তব্য IP ঠিকানা দেখে forwarding করে।

এটি forwarding table অনুযায়ী কাজ করে, এবং প্রয়োজনে Longest Prefix Match ব্যবহার করে সঠিক রুট নির্বাচন করে।

Longest prefix matching

longest prefix match

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011** **	2
otherwise	3

examples: 11001000 00010111 00010110 10100001 which interface?
11001000 00010111 00011000 10101010 which interface?

🌐 Longest Prefix Matching (LPM) কী?

যখন একটি রাউটার কোনো প্যাকেট পায়,
তখন সেটি প্যাকেটের **Destination IP Address** দেখে forwarding table-এ খুঁজে দেখে
কোন network prefix (বা range) তার সাথে মিলে।

কিন্তু—

একই IP address একাধিক রেঞ্জ বা prefix-এর সাথে মিলে যেতে পারে।
এই ক্ষেত্রে, রাউটার বেছে নেয় যে prefix সবচেয়ে বেশি বিটের জন্য মিলে,
অর্থাৎ সবচেয়ে নির্দিষ্ট route।

৩ সহজভাবে বললে:

“যে route সবচেয়ে সুনির্দিষ্ট (longest match), সেটাই বেছে নেওয়া হয়।”

□ Forwarding Table Example

Destination Address Prefix	Link Interface
11001000 00010111 00010*** ***** 0	
11001000 00010111 0001100* ***** 1	
11001000 00010111 00011*** ***** 2	
Otherwise	3

❖ Rule of Longest Prefix Match

- প্রতিটি রেঞ্জের prefix-এর দৈর্ঘ্য (যত বিট নির্দিষ্ট) গুরুত্বপূর্ণ।
 - যত দীর্ঘ prefix, তত নির্দিষ্ট route।
 - একই IP একাধিক prefix-এ পড়লে, রাউটার **longest prefix** নির্বাচন করে।
-

🔍 Example Analysis

Example 1:

Destination Address →

11001000 00010111 00010110 10100001

এখন দেখা যাক, এটি কোন prefix-এর সাথে মেলে ছে

Prefix	Matches?	Bits Matched	Interface
11001000 00010111 00010*** *****	✓	19 bits	0
11001000 00010111 0001100* *****	✗	—	1
11001000 00010111 00011*** *****	✓	18 bits	2

- এখানে prefix 1 (interface 0) সবচেয়ে বেশি বিটে মিলে গেছে (19 bits)
✓ Forward to Interface 0
-

Example 2:

Destination Address →

11001000 00010111 00011100 10101010

Prefix	Matches?	Bits Matched	Interface
11001000 00010111 00010*** *****	✗	—	0
11001000 00010111 0001100* *****	✗	—	1
11001000 00010111 00011*** *****	✓	19 bits	2

- এই ক্ষেত্রে prefix 3 (interface 2) সবচেয়ে বেশি বিটে মিলে গেছে।
✓ Forward to Interface 2
-

ঔষধ Longest Prefix Matching-এর মূল উদ্দেশ্য

লক্ষ্য	ব্যাখ্যা
সুনির্দিষ্ট Routing	প্যাকেটকে সর্বাধিক নির্দিষ্ট (specific) subnet-এ পাঠানো
Efficient Routing Table	বড় নেটওয়ার্কে route কমিয়ে আনতে সাহায্য করে
Flexibility	নেটওয়ার্ক subdivision (subnetting) সহজ করে তোলে

□ সংক্ষেপে

ধাপ	কাজ
①	প্যাকেট আসে ইনপুট পোর্টে

ধাপ	কাজ
২	Destination IP Address দেখা হয়
৩	Forwarding Table-এর প্রতিটি prefix-এর সাথে তুলনা করা হয়
৪	যে prefix সবচেয়ে বেশি বিটে মেলে, সেটি বেছে নেওয়া হয়
৫	প্যাকেটটি এই prefix-এর Link Interface দিয়ে পাঠানো হয়

✓ মূল ধারণা

Longest Prefix Matching (LPM) হল রাউটার-এর forwarding কৌশল, যেখানে একটি প্যাকেটের destination IP address-এর সাথে সবচেয়ে নির্দিষ্ট prefix (longest match) খুঁজে বের করে সেটির ভিত্তিতে forwarding সিদ্ধান্ত নেওয়া হয়।

🌐 DHCP (Dynamic Host Configuration Protocol)

❖ Goal / উদ্দেশ্য:

DHCP এমন একটি প্রোটোকল যা host বা device-কে স্বয়ংক্রিয়ভাবে IP address দেয় যখন সেটি নেটওয়ার্কে যুক্ত (join) হয়।

৩ সহজভাবে বললে —

“DHCP হলো নেটওয়ার্কের এমন একটি সেবা,
যা তোমার কম্পিউটার বা ফোনকে স্বয়ংক্রিয়ভাবে IP address, gateway, এবং DNS দেয়।”

❖ DHCP-এর মূল লক্ষ্যসমূহ

- ▢ **Dynamic allocation:** Host যখন নেটওয়ার্কে আসে, DHCP server তাকে একটি IP address দেয়।
- ☒ **Lease renewal:** Host তার IP address নির্দিষ্ট সময়ের জন্য lease হিসেবে পায়, এবং প্রয়োজন হলে সেটি renew করতে পারে।
- ☒ **Reuse of addresses:** একটি IP address তখনই ব্যবহৃত হয় যখন host অনলাইনে থাকে; disconnect হলে address আবার অন্য host-কে দেয়া যায়।

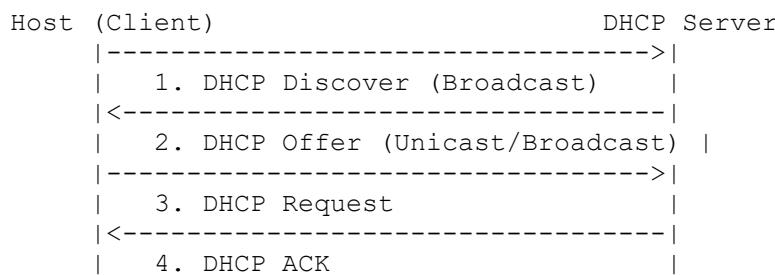
- **Mobile user support:**
যারা প্রায়ই নেটওয়ার্ক পরিবর্তন করে (যেমন ল্যাপটপ, ফোন),
তারা প্রতিবার নতুন IP address স্বয়ংক্রিয়ভাবে পায়।

পঞ্চ DHCPOverview — Message Exchange Process

DHCP সাধারণত চারটি মূল ধাপে কাজ করে ছে
(এটিকে প্রায়ই বলা হয় DORA process — Discover, Offer, Request, Acknowledge)

ধাপ	বার্তা (Message)	প্রেরক → গ্রাহক	বর্ণনা
1	DHCP Discover	Host → Broadcast	Host জানায় “আমি নেটওয়ার্কে আছি, কেউ কি আমাকে IP দিতে পারবে?”
2	DHCP Offer	Server → Host	DHCP Server একটি address প্রস্তাব করে (যেমন 192.168.1.10)।
3	DHCP Request	Host → Server	Host বলে “আমি ওই IP address নিতে চাই।”
4	DHCP ACK (Acknowledge)	Server → Host	Server আনুষ্ঠানিকভাবে সেই IP address প্রদান করে এবং Lease সময় জানায়।

□ DHCP Message Flow Diagram (সহজভাবে):



পঞ্চ DHCP-এর গুরুত্বপূর্ণ তথ্যসমূহ

DHCP Server কেবল IP ঠিকানা দেয় না,
এটি host-কে আরও কিছু গুরুত্বপূর্ণ নেটওয়ার্ক তথ্যও প্রদান করে ছে

তথ্য উদাহরণ

IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Default Gateway	192.168.1.1
DNS Server	8.8.8.8, 8.8.4.4
Lease Time	24 hours

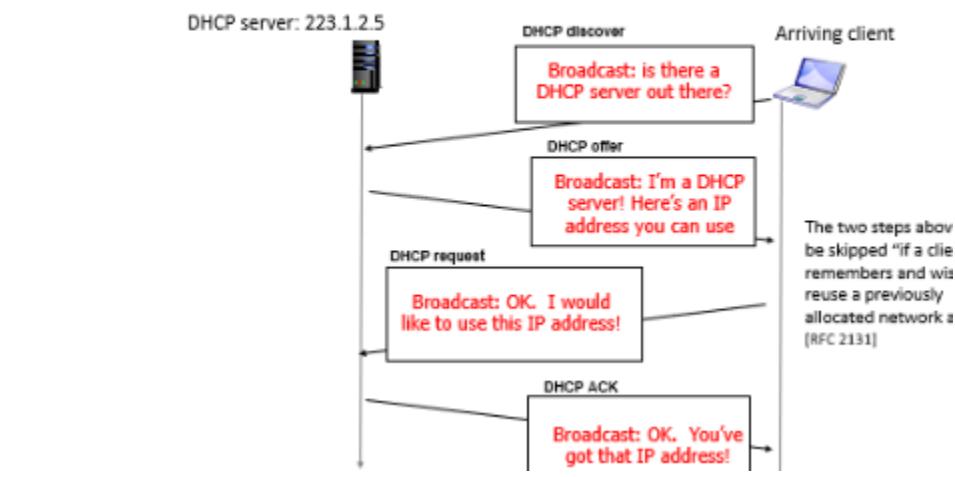
□ DHCP-এর সুবিধাসমূহ

সুবিধা	ব্যাখ্যা
Automation	IP বরাদ্দের কাজ স্বয়ংক্রিয় হয়, কোনো manual setup লাগে না।
Efficiency	IP reuse করে address space সংরক্ষণ করে।
Mobility	মোবাইল বা ল্যাপটপ ব্যবহারকারীদের জন্য উপযোগী।
Error Reduction	হাতে টাইপ করা ভুল (যেমন duplicate IP) এড়ানো যায়।

✓ মূল ধারণা

DHCP (Dynamic Host Configuration Protocol) host-কে dynamicভাবে IP address দেয় এবং নেটওয়ার্ক কনফিগারেশন স্বয়ংক্রিয় করে — ফলে নেটওয়ার্ক ব্যবস্থাপনা সহজ, কার্যকর, এবং ভুলমুক্ত হয়।

DHCP client-server scenario



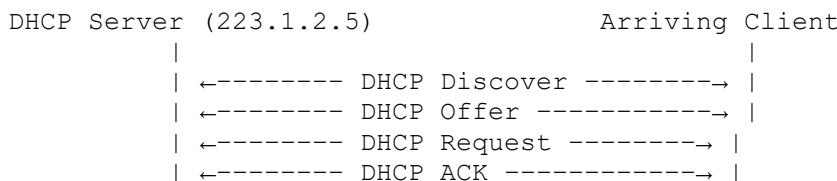
🌐 DHCP Client–Server Scenario (DHCP কার্যপ্রবাহ)

ধরা যাক,
একটি নতুন কম্পিউটার (client) নেটওয়ার্কে সংযুক্ত হলো —
এখন তার কোনো IP address নেই।
তখন DHCP প্রক্রিয়া শুরু হয়।

✿ চারটি প্রধান ধাপ (DORA process)

ধাপ	মেসেজ নাম	প্রেরক → গ্রাহক	বর্ণনা
①	DHCP Discover	Client → Broadcast	ক্লায়েন্ট জিজেস করে: “Is there any DHCP server out there?” (অর্থাৎ, কেউ কি আমাকে IP দিতে পারবে?)
②	DHCP Offer	Server → Broadcast	DHCP Server উত্তর দেয়: “আমি DHCP Server, এই নাও একটি IP address (যেমন 192.168.1.10)”
③	DHCP Request	Client → Broadcast	ক্লায়েন্ট বলে: “আমি ওই IP address ব্যবহার করতে চাই।”
④	DHCP ACK	Server → Broadcast	DHCP Server নিশ্চিত করে: “ঠিক আছে, এখন থেকে এই IP address তোমার।”

□ Message Flow (চিত্রে যেমন দেখা যাচ্ছে):



⦿ প্রতিটি মেসেজের ব্যাখ্যা

□ 1. DHCP Discover

Broadcast: “Is there a DHCP server out there?”

✿ ক্লায়েন্ট নেটওয়ার্কে ব্রডকাস্ট করে জানায় যে সে IP address চায়।
এটি একটি broadcast message, কারণ ক্লায়েন্ট এখনো জানে না সার্ভার কোথায়।

□ 2. DHCP Offer

Broadcast: “I’m a DHCP server! Here’s an IP address you can use.”

ଡିଜାଲ୍ ସାର୍ଭାର ଉତ୍ତର ଦେଇ ଏବଂ ଏକଟି **অস্থায়ী IP address** ପ୍ରସ୍ତାବ କରେ,
ସାଥେ subnet mask, gateway, DNS ଏବଂ lease time ଦେଇ।

● 3. DHCP Request

Broadcast: “OK, I would like to use this IP address!”

ଡିଜାଲ୍ କ୍ଲାଯେନ୍ଟ ବଲେ ଯେ ସେ ଓଇ ପ୍ରସ୍ତାବିତ IP address ଗ୍ରହଣ କରାଚେ।
ଯଦି ଏକାଧିକ ସାର୍ଭାର offer ପାଠାଯ, କ୍ଲାଯେନ୍ଟ ଏକଟିରଟି ବେଛେ ନେଇ ଏବଂ ବାକିଦେର ଉପେକ୍ଷା କରେ।

● 4. DHCP ACK (Acknowledge)

Broadcast: “OK, you’ve got that IP address!”

ଡିଜାଲ୍ ସାର୍ଭାର ଆନୁଷ୍ଠାନିକଭାବେ ଜାନାଯ ଯେ
ଏହି IP address ଏଥିନେ କ୍ଲାଯେନ୍ଟରେ ଜନ୍ୟ lease (ଭାଡାଯ) ବରାଦ କରା ହେବେ।

▣ Lease Renewal ଓ Reuse

- DHCP ସାର୍ଭାର ପ୍ରତିଟି IP address ନିର୍ଦ୍ଦିଷ୍ଟ ସମୟର ଜନ୍ୟ ଦେଇ (ଯେମନ 24 ଘଣ୍ଟା)।
 - ଏହି ସମୟ ଶେଷ ହୋଇଲୁବାର ଆଗେ କ୍ଲାଯେନ୍ଟ DHCP Request ପାଠିଯେ lease renew କରାନ୍ତେ ପାରେ।
 - ଯଦି କ୍ଲାଯେନ୍ଟ ନେଟୋୱାର୍କ ଛେଡେ ଯାଇ (disconnect), ତାହାରେ ଏହି IP ଆବାର ଅନ୍ୟ host-କେ ଦେଇ ଯାଇ।
-

▣ Optional Step: Reconnection Optimization

ପ୍ଲାଇଡେର ଡାନପାଶେ ଯେ ଲେଖା ଆଛେ —

“The two steps above can be skipped if a client remembers and still can reuse a previously allocated address.”

এর মানে হলো

যদি কোনো ক্লায়েন্ট আগে DHCP থেকে একটা IP পেয়ে থাকে এবং এখন আবার একই

নেটওয়ার্কে ফিরে আসে,

তাহলে Discover/Offer ধাপগুলো বাদ দিয়ে সরাসরি Request/ACK ধাপে চলে যেতে পারে।

সংক্ষেপে মূল ধারণা

বৈশিষ্ট্য

DHCP কীভাবে কাজ করে

Automation IP address, subnet mask, gateway ইত্যাদি স্বয়ংক্রিয়ভাবে দেয়

Broadcast প্রাথমিক পর্যায়ে broadcast ব্যবহার করে, কারণ ক্লায়েন্টের IP নেই

Temporary Lease প্রতিটি IP নির্দিষ্ট সময়ের জন্য দেওয়া হয়

Reuse একই IP পরে অন্য host ব্যবহার করতে পারে

Renewal Host চাইলে lease নবায়ন করতে পারে

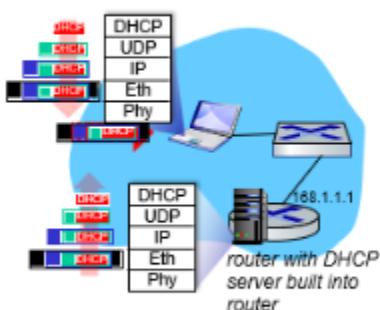
উদাহরণ:

নতুন ল্যাপটপ নেটওয়ার্কে যুক্ত হলে DHCP প্রক্রিয়া শুরু হয়

→ DHCP সার্ভার তাকে IP দেয়

→ এখন ল্যাপটপে ইন্টারনেট অ্যাক্সেস করা যায়

DHCP: example



- Connecting laptop will use DHCP to get IP address, address of first-hop router, address of DNS server.
 - DHCP REQUEST message encapsulated in UDP, encapsulated in IP, encapsulated in Ethernet
 - Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server
 - Ethernet de-mux'ed to IP de-mux'ed, UDP de-mux'ed to DHCP
-

DHCP Example: বাস্তবে কিভাবে কাজ করে

ধরা যাক, তুমি তোমার ল্যাপটপ নিয়ে Wi-Fi রাউটারে যুক্ত হলে।

রাউটারটি DHCP সার্ভার হিসেবেও কাজ করছে (built-in DHCP service)।

তখন DHCP প্রোটোকলের মাধ্যমে তোমার ল্যাপটপ তার IP ও অন্যান্য নেটওয়ার্ক তথ্য পায়।

শুষ্ক ধাপ-ধাপে ব্যাখ্যা

1 Connecting Laptop

- ল্যাপটপ নেটওয়ার্কে যুক্ত হওয়ার সময় **DHCP Discover** বার্তা পাঠায়।
- এর উদ্দেশ্য হলো — “আমাকে একটি IP address দাও।”
- DHCP সার্ভার (রাউটার) এই অনুরোধ গ্রহণ করে।

৩ DHCP সার্ভার ল্যাপটপকে দেয়:

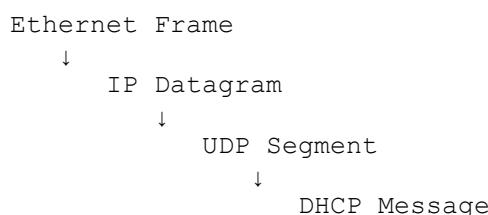
- IP address (যেমন **192.168.1.10**)
 - First-hop router address (gateway) — **192.168.1.1**
 - DNS server address (যেমন **8.8.8.8**)
 - Subnet mask (যেমন **255.255.255.0**)
-

2 Message Encapsulation (বার্তা প্যাকেটাইজেশন)

DHCP বার্তাগুলি অনেক স্তরে encapsulate হয়ে নেটওয়ার্কে পাঠানো হয় ক্ষ

প্রোটোকল স্তর	কাজ
DHCP	মেসেজ তৈরি করে (Request, Offer, ACK ইত্যাদি)
UDP	DHCP মেসেজ UDP ডাটাগ্রামে encapsulate করে (port 67–68)
IP	UDP প্যাকেটকে IP datagram হিসেবে encapsulate করে
Ethernet	IP datagram-কে Ethernet frame-এ প্যাকেট করে

ক্ষ সুতরাং, DHCP Request প্যাকেটটি দেখতে হবে এভাবে:



৩ Broadcast Communication

- যেহেতু ল্যাপটপের এখনও কোনো IP নেই, DHCP Discover বার্তা পাঠানো হয় broadcast address (FF:FF:FF:FF:FF:FF)-এ।
- রাউটার (যার মধ্যে DHCP server built-in আছে) LAN-এর broadcast শুনে বার্তাটি গ্রহণ করে এবং প্রতিক্রিয়া জানায়।

৪ De-multiplexing Process (De-mux)

প্রাপ্ত প্যাকেট রাউটারে ধাপে ধাপে "খোলা" (de-encapsulate) হয় ক্ষেত্রে

ধাপ	ব্যাখ্যা
Ethernet de-mux	ফ্রেম থেকে IP datagram আলাদা করা হয়।
IP de-mux	IP থেকে UDP segment বের করা হয়।
UDP de-mux	UDP থেকে DHCP message বের করে DHCP প্রোটোকলে পাঠানো হয়।

এভাবেই DHCP সার্ভার প্যাকেটের ভেতরের আসল DHCP অনুরোধ বার্তাটি পায়।

৫ DHCP শুধু IP address দেয় না — আরও তথ্য দেয়

DHCP ক্লায়েন্টকে শুধু IP address নয়, বরং পূর্ণ নেটওয়ার্ক কনফিগারেশন দেয় ক্ষেত্রে

তথ্য	উদাহরণ	অর্থ
IP Address	192.168.1.10	ক্লায়েন্টের নিজস্ব ঠিকানা
Subnet Mask	255.255.255.0	নেটওয়ার্ক ও হোস্ট অংশ আলাদা করে
Default Gateway	192.168.1.1	প্রথম হপ রাউটার, যেখানে প্যাকেট যাবে
DNS Server	8.8.8.8, 8.8.4.4	ডোমেইন নাম রেজল্ভ করার জন্য
Lease Time	24 hours	এই IP ক্রতৃপক্ষের অনুমতি আছে

৬ উদাহরণ পরিস্থিতি

তোমার ল্যাপটপ নেটওয়ার্কে সংযুক্ত হলো
DHCP সার্ভার (রাউটার) বলল:
“তুমি এখন 192.168.1.10 ব্যবহার করো, আর আমি তোমার গেটওয়ে 192.168.1.1।”

এখন ল্যাপটপ ইন্টারনেটে যেতে পারবে
কারণ সে জানে কে তার gateway, এবং কোন DNS সার্ভার ব্যবহার করতে হবে।

✓ সংক্ষেপে মূল ধারণা

বিষয়	ব্যাখ্যা
DHCP কাজ করে Client–Server মডেল	ক্লায়েন্ট অনুরোধ করে, সার্ভার IP বরাদ্দ করে
Encapsulation স্তরভিত্তিক	DHCP → UDP → IP → Ethernet
Broadcast ব্যবহৃত হয়	ক্লায়েন্টের কোনো IP না থাকায়
DHCP সার্ভার সম্পূর্ণ কনফিগারেশন দেয় শুধু IP নয়, Gateway, DNS, Mask সহ	

IP addresses: how to get one?

Q: how does *network* get subnet part of IP address?

A: gets allocated portion of its provider ISP's address space

ISP's block 11001000 00010111 00010000 00000000 200.23.16.0/20

ISP can then allocate out its address space in 8 blocks:

Organization 0	<u>11001000 00010111 00010000 00000000</u>	200.23.16.0/23
Organization 1	<u>11001000 00010111 00010010 00000000</u>	200.23.18.0/23
Organization 2	<u>11001000 00010111 00010100 00000000</u>	200.23.20.0/23
...
Organization 7	<u>11001000 00010111 00011110 00000000</u>	200.23.30.0/23

🌐 IP Addresses: How to Get One?

? প্রশ্ন:

কিভাবে একটি **network** তার subnet অংশের IP address পায়?

💡 উত্তর:

এটি তার provider ISP-এর IP address space থেকে একটি portion (subnet) বরাদ্দ পায়।

□ Step 1: ISP-এর বড় Address Block

ধরা যাক, একটি ISP (যেমন BTCL, ISP কোম্পানি ইত্যাদি)
একটি বড় IP address ব্লক পেয়েছে —

ISP's block:

200.23.16.0/20

এর অর্থ:

- Network prefix = /20
 - মানে প্রথম 20 bits নেটওয়ার্ক অংশ
 - বাকি 12 bits (32 - 20) host অংশ
 - অর্থাৎ মোট address সংখ্যা = $2^{12} = 4096$ টি IP address
-

□ Step 2: ISP এটি ভাগ করবে ছোট ছোট Subnet-এ

এখন ISP চায় এই বড় ব্লককে ভাগ করে 8টি ছোট ব্লকে বিভিন্ন organization-কে দিতে।

8 ভাগ করতে হলে দরকার অতিরিক্ত 3 bits (কারণ $2^3 = 8$)।

অর্থাৎ, নতুন prefix হবে /23 (20 + 3)।

৩ তাই প্রতিটি সাবনেট হবে /23 —

অর্থাৎ প্রতি সাবনেটে থাকবে $2^9 = 512$ টি IP address।

ঃ Step 3: Allocation Table

Organization	Binary Address (Network Portion)	Subnet Mask	CIDR Notation
Org 0	11001000 00010111 00010000 00000000	/23	200.23.16.0/23
Org 1	11001000 00010111 00010010 00000000	/23	200.23.18.0/23
Org 2	11001000 00010111 00010100 00000000	/23	200.23.20.0/23
...
Org 7	11001000 00010111 00011110 00000000	/23	200.23.30.0/23

Step 4: ফলাফল (Result)

ISP তার বড় address block /20 কে ভাগ করেছে ৪টি /23 সাবনেটে।

আইটেম	মান
মূল ব্লক	200.23.16.0/20
মোট IP সংখ্যা	4096
সাবনেট সংখ্যা	8
প্রতিটি সাবনেটের IP সংখ্যা	512
প্রতিটি সাবনেটের Prefix	/23

সহজভাবে বোঝো

ধরা যাক, ISP একটি বড় জমি পেয়েছে (200.23.16.0/20) এখন সেই জমি ৪টি কোম্পানিকে সমান ভাগে দিচ্ছে।
প্রতিটি কোম্পানির জন্য নির্দিষ্ট একটি ব্লক বরাদ্দ (যেমন /23)।
এই ব্লকই হলো তাদের subnet address space।

সংক্ষেপে সারাংশ

ধারণা	ব্যাখ্যা
ISP block	ISP-এর বড় address space (যেমন 200.23.16.0/20)
Subnetting	এই ব্লককে ছোট অংশে ভাগ করা (/23, /24 ইত্যাদি)
Allocation	প্রতিটি অংশ নির্দিষ্ট organization-কে বরাদ্দ করা
CIDR notation	Classless Inter-Domain Routing — যেমন 200.23.16.0/23
Prefix length	ষত বড় prefix (যেমন /28), তত ছোট subnet

মূল ধারণা

একটি প্রতিষ্ঠান (network) তার IP subnet ISP-এর address block থেকেই পায়।
ISP তার বড় address space (যেমন /20) ছোট subnet (/23) আকারে ভাগ করে
বিভিন্ন গ্রাহক (organization)-কে বরাদ্দ করে।

