FORE
SYSTEMS

IPv6: Why Do We Need A New IP?

V1.2: Geoff Bennett

IPv4:
Today's IP

Version 4 of the IP protocol (IPv4) has been used very successfully to build the largest internetworks in the world. Clearly we need to have very good reasons to change.

IP Version 4

IPv4 has certain characteristics that have made it very successful...

IP Version 4

- **User-Driven Standard**

IP has always been a user-driven standard. This is in sharp contrast to OSI protocols like the Connectionless Network Layer Protocol (CLNP) which is defined by an international standards body, ISO.

IP Version 4

- User-Driven Standard
- **Hierarchical Address Space**

IP Address = Network_ID . Host_ID

IP has a hierarchical address space. Initially this was conceived to be a two-tier hierarchy of Network ID and Host ID.

IP Version 4

- User-Driven Standard
- **Hierarchical Address Space**

IP Address = Network_ID . Host_ID

IP Address = Network_ID . Subnet_ID . Host_ID

Later this hierarchy was extended as described in RFC 950 to include a subnet tier.

IP Version 4

- User-Driven Standard
- Hierarchical Address Space
- **Centralised Address Allocation**

Centralised address allocation means that a registered IP address is guaranteed to be unique throughout the Internet. Similar to a telephone number.

IP Version 4

- User-Driven Standard
- Hierarchical Address Space
- Centralised Address Allocation
- **Autonomous System Concept**

Finally, the Internet architecture included the concept of the Autonomous System. This feature allows IP internetworks to be built under the control of different design and administration groups, without risk of interference.

IPv4 Limitations

Considering that the current IP RFC (791) dates from 1981, IPv4 has been incredibly persistent. However, there are certain major limitations that exist in the protocol today.

- **Pressure on Class B Address Space**
- **Routing Table Size**
- **Overall Address Exhaustion**
- **Other Issues**

Three big limitations dominate the drive for an updated IP.
I’ve listed these in order of urgency, although the first two items are essentially tied for first place.
Other limitations are much less important, but if we’re thinking of changing IP, we may as well solve these problems too.

IPV4: Address Space Pressure

- **Class Size Limitations**

IPV4: Address Space Pressure

- **Class Size Limitations**

Class A : 127 Networks of 16.8 Million Hosts

Class B : 8k Networks of 65k Hosts

Class C : 2 Million Networks of 254 Hosts

IP address classes are very restrictive for the way we like to build networks today.

IPV4: Address Space Pressure

- **Class Size Limitations**

Class A : 127 Networks of 16.8 Million Hosts

7 bits **24 bits**

Net_ID . **Host_ID**

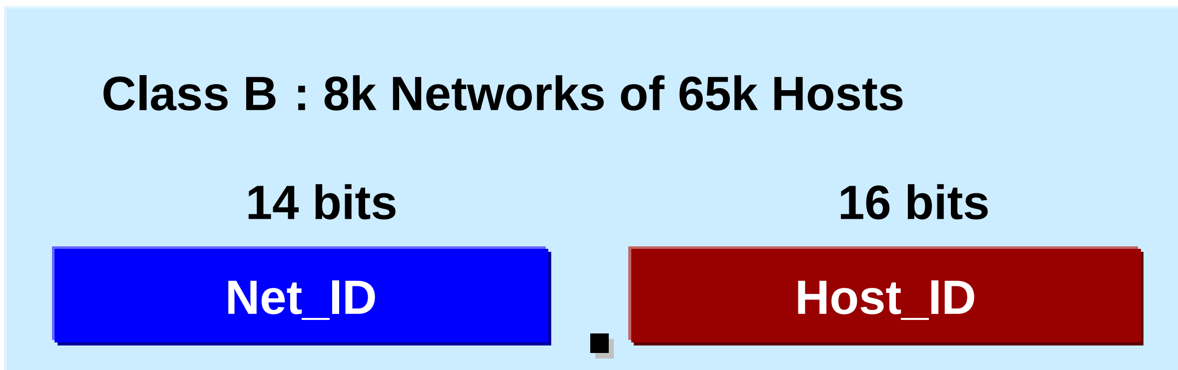
Of course we would all like to be allocated a Class A address, which leaves plenty of scope in the Host ID field for efficient subnetting.

Unfortunately only 127 networks in the world can be allocated a Class A address.

And one of these - 127.0.0.0 - is reserved for loopback!

IPV4: Address Space Pressure

- **Class Size Limitations**



Class B networks are the next best thing. The Host ID field is big enough to allow quite flexible subnetting. However, with only eight thousand or so addresses available, Class B address space is at a premium. Over half of the total Class B address space has been allocated, and the NIC is very reluctant to provide these addresses.

IPV4: Address Space Pressure

- **Class Size Limitations**

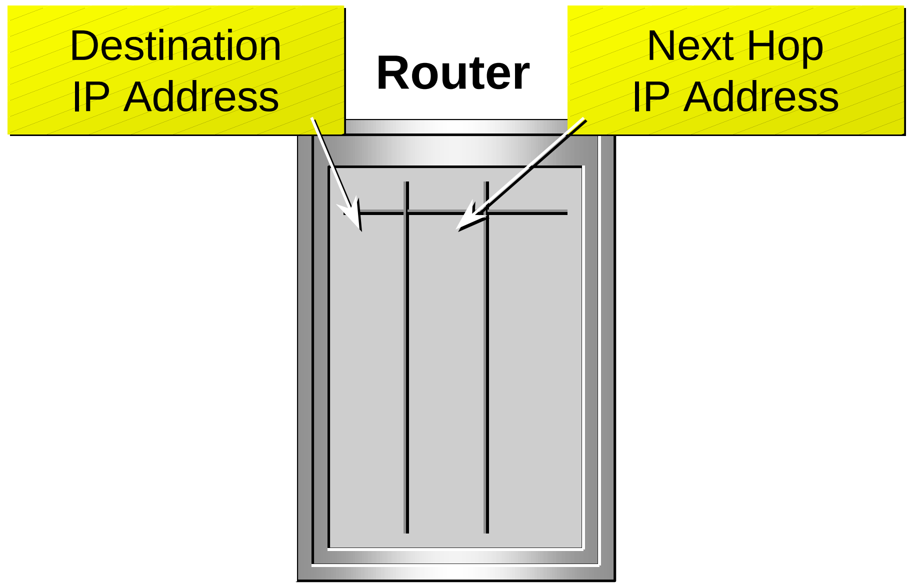
Class C : 2 Million Networks of 254 Hosts

21 bits		8 bits
Net_ID	.	Host_ID

Class C networks would appear to be ideal. There are plenty of addresses, and each Network ID contains enough Host ID numbers for most users.

However, there is a fundamental problem with the scalability of Class C address space.

IPV4: Routing Table Size

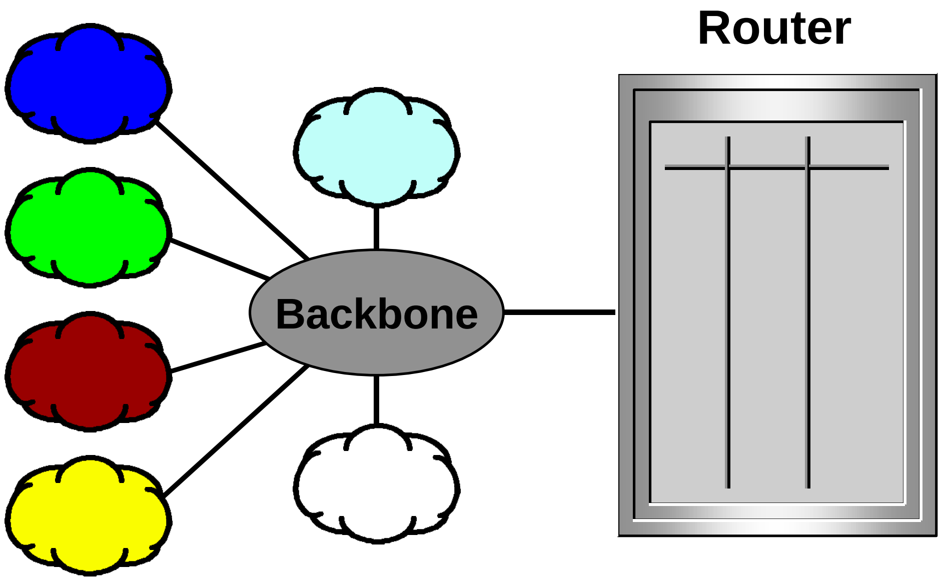


Inside every IP router is a Routing Table.

This table contains a list of destination IP addresses, with corresponding “Next Hop” IP addresses.

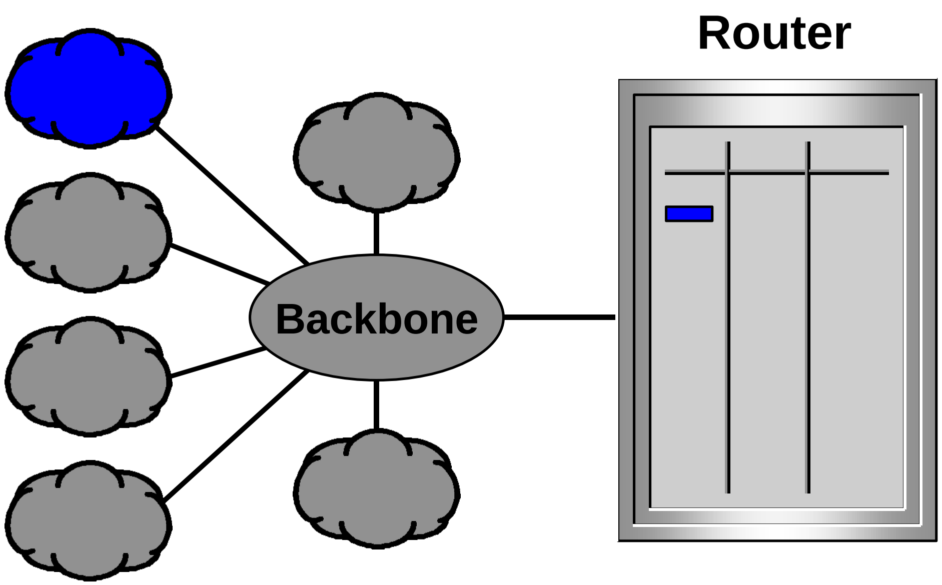
There are also columns for costing the route, the type of routing protocol used, etc. These columns aren’t really relevant to this discussion so I’ll leave them out.

IPV4: Routing Table Size



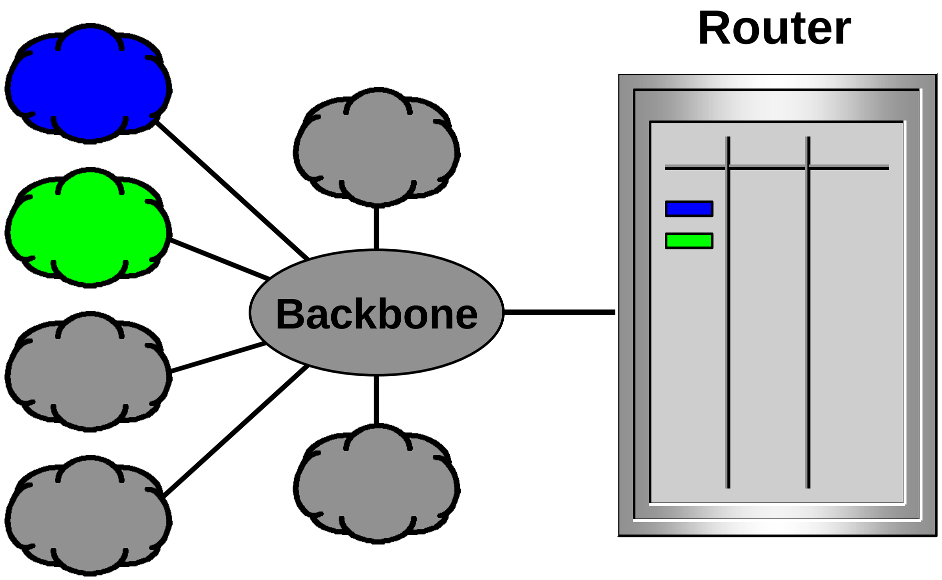
In this example, let's say that this router is connected into a Corporate Internetwork with many IP networks attached to a backbone.

IPV4: Routing Table Size



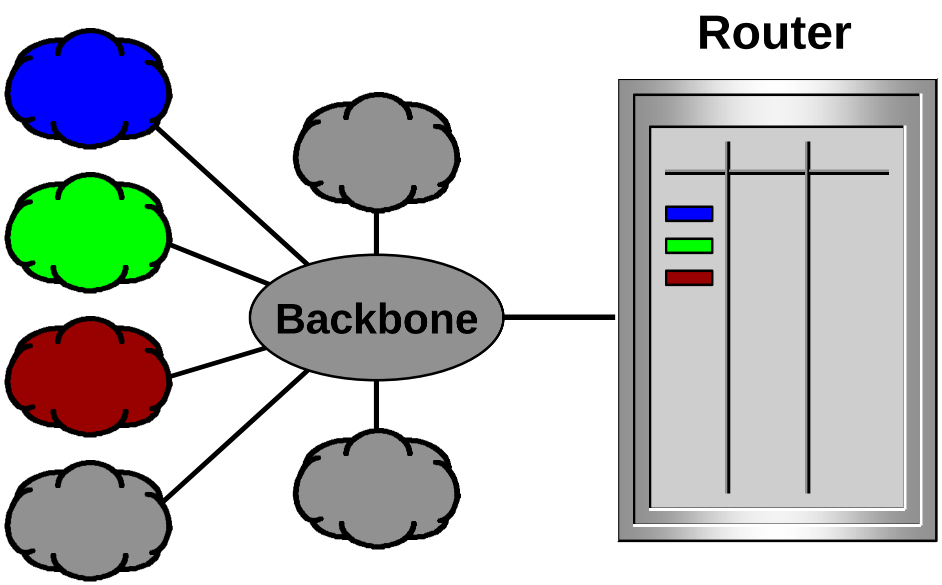
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



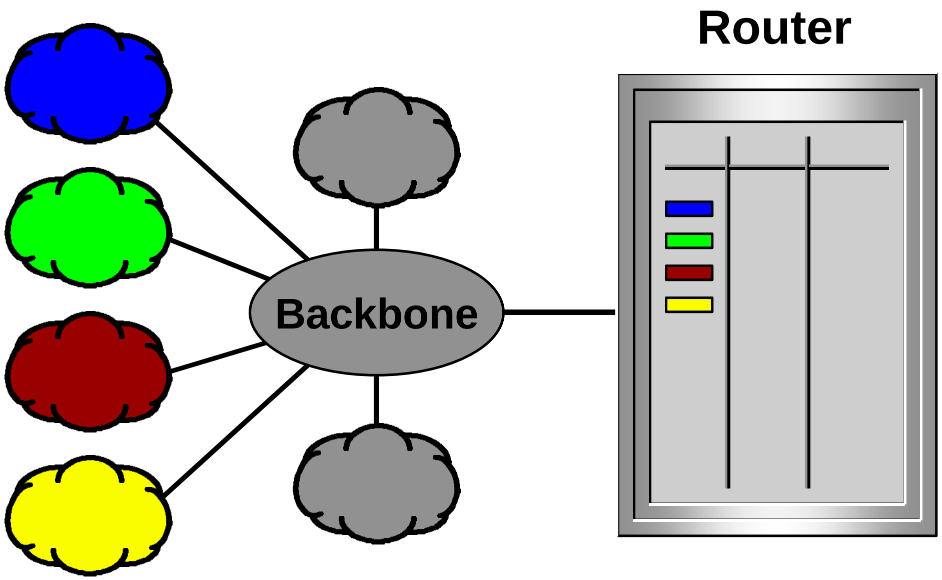
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



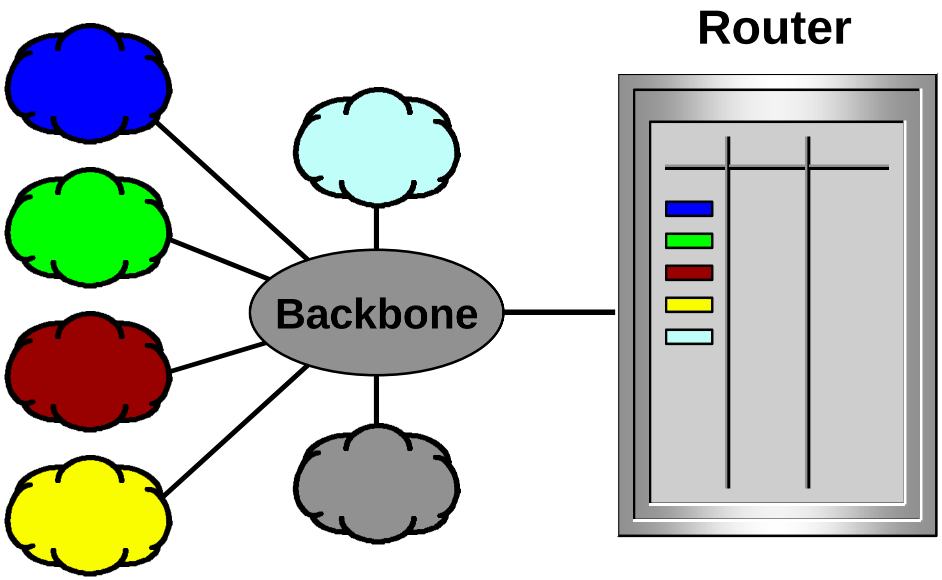
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



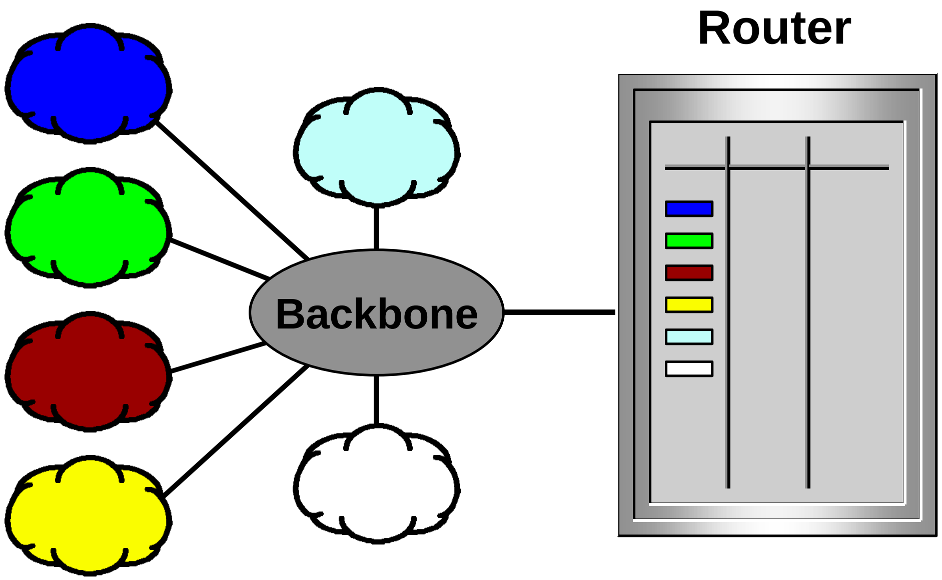
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



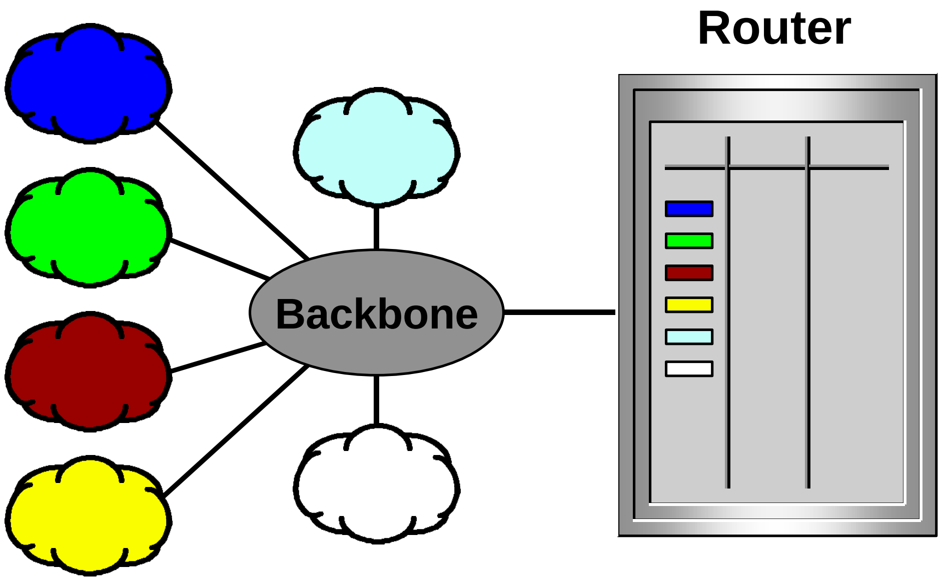
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



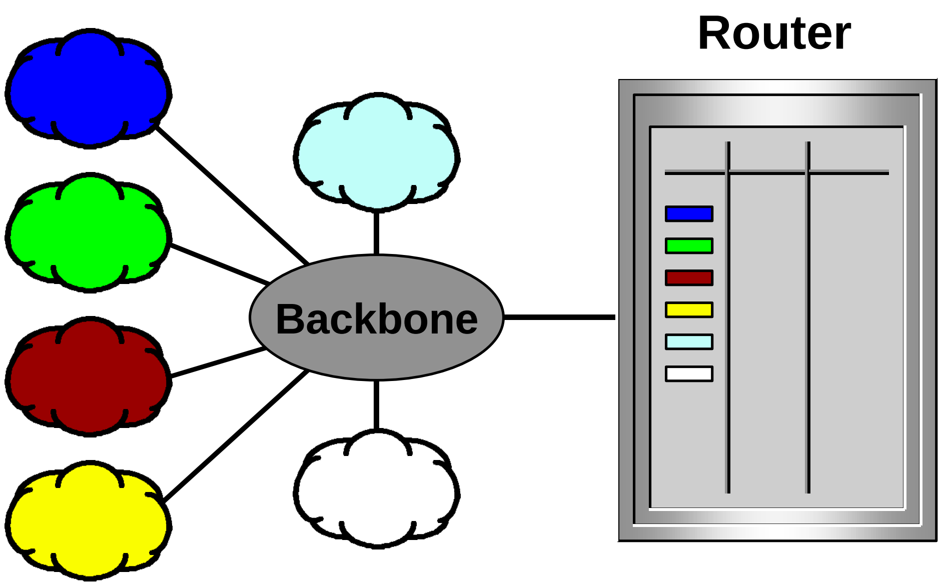
One Routing Table entry is required for each IP network that is discovered by the router.

IPV4: Routing Table Size



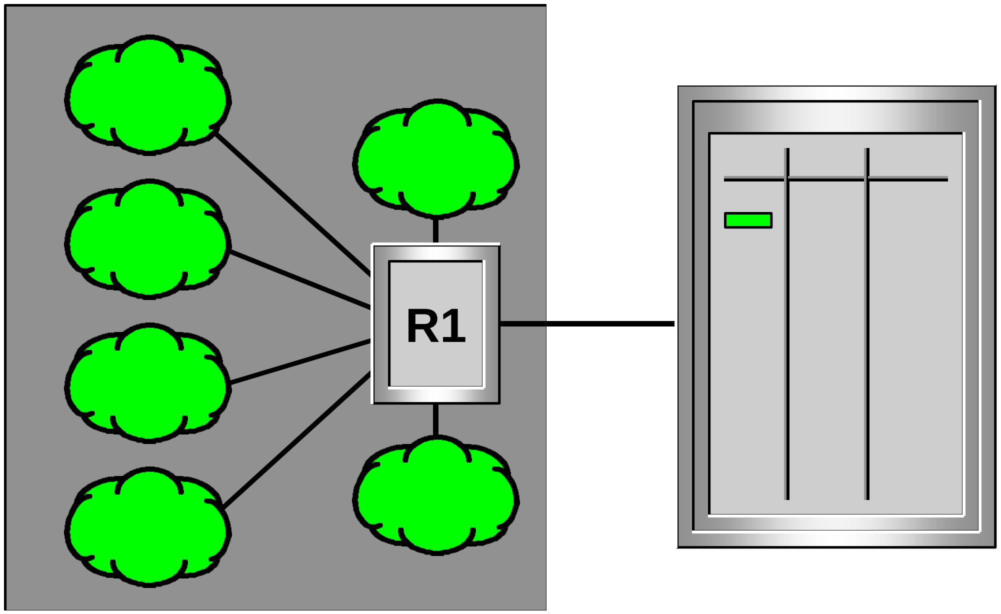
One Routing Table entry is required for each IP network that is discovered by the router.
Fortunately in private networks, the size of Routing Tables is unlikely to grow beyond a manageable level.

IPV4: Routing Table Size



However, if we're able to use subnetting, we can "hide" the network structure and economise on routing table space.
To do this, we need to look at the network design from a slightly different angle...

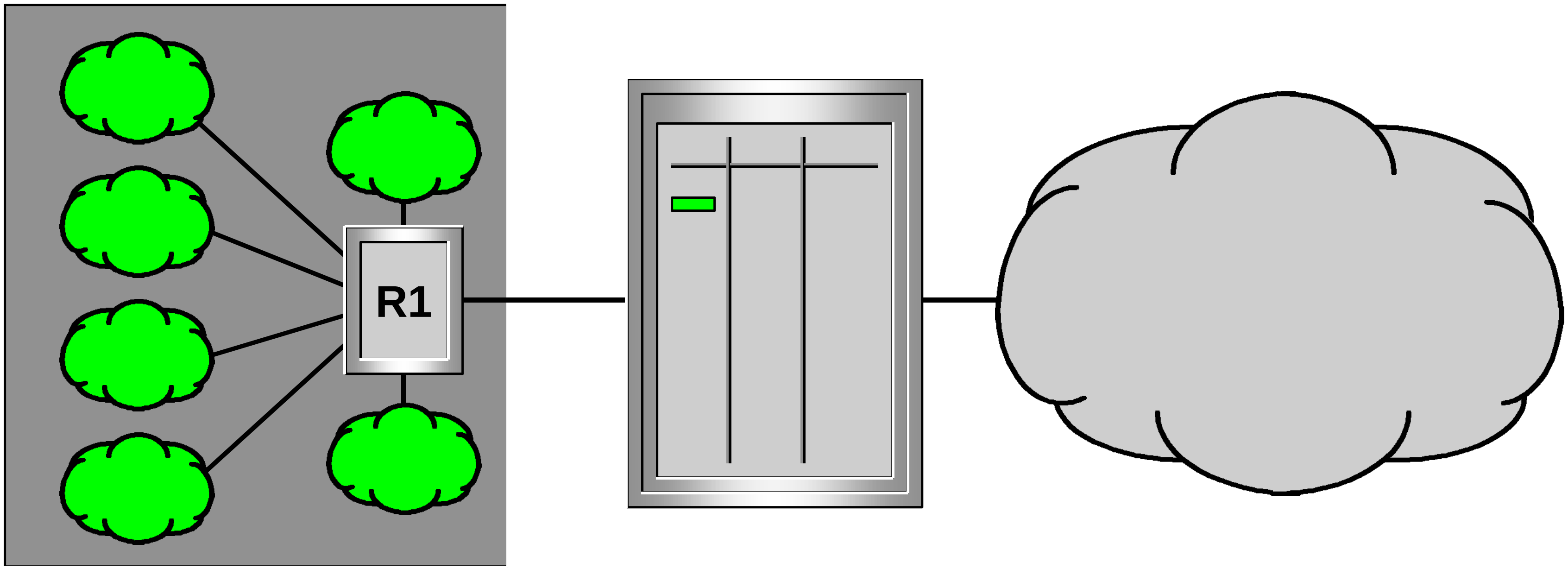
IPV4: Routing Table Size



In this design, if all the networks are subnets of the same Network ID, then a single entry is all that is required. Note that "upstream" routers such as R1 in this diagram must be used to perform subnet agglomeration.

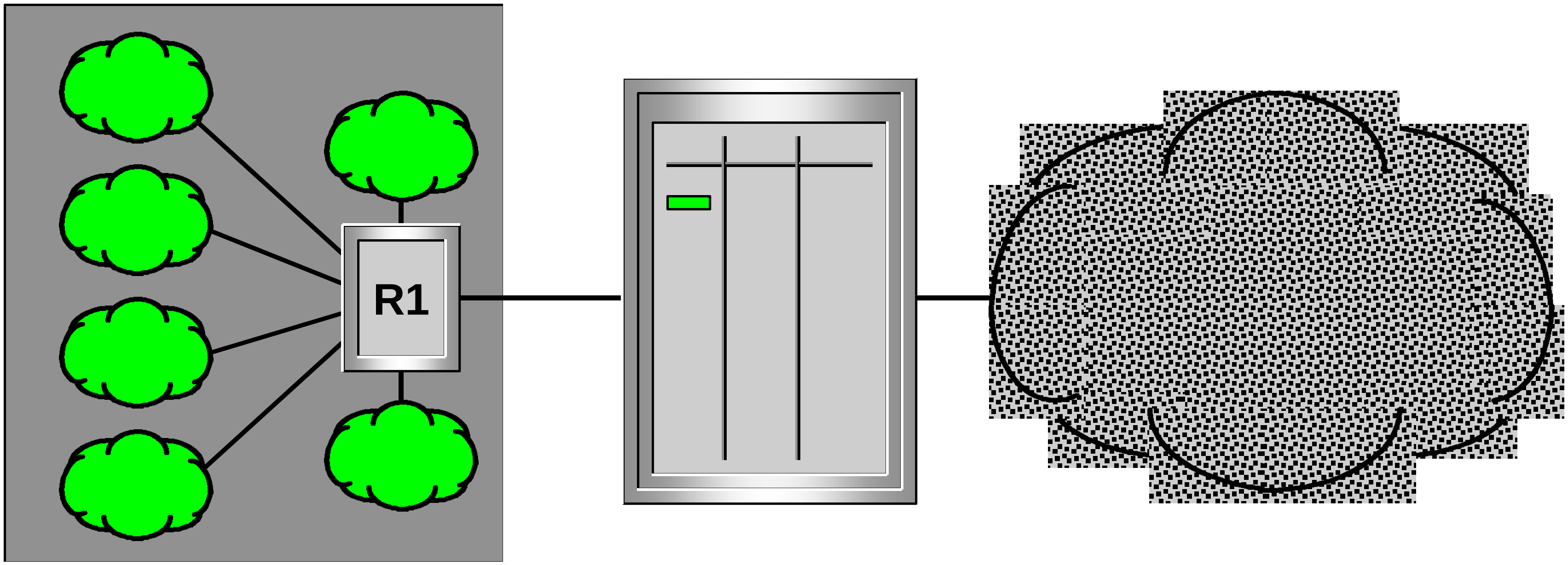
While subnetting is very easy with Class A or B addresses, it's more difficult with a Class C address because we have to subnet within a byte boundary and this means thinking in BINARY.

IPV4: Routing Table Size



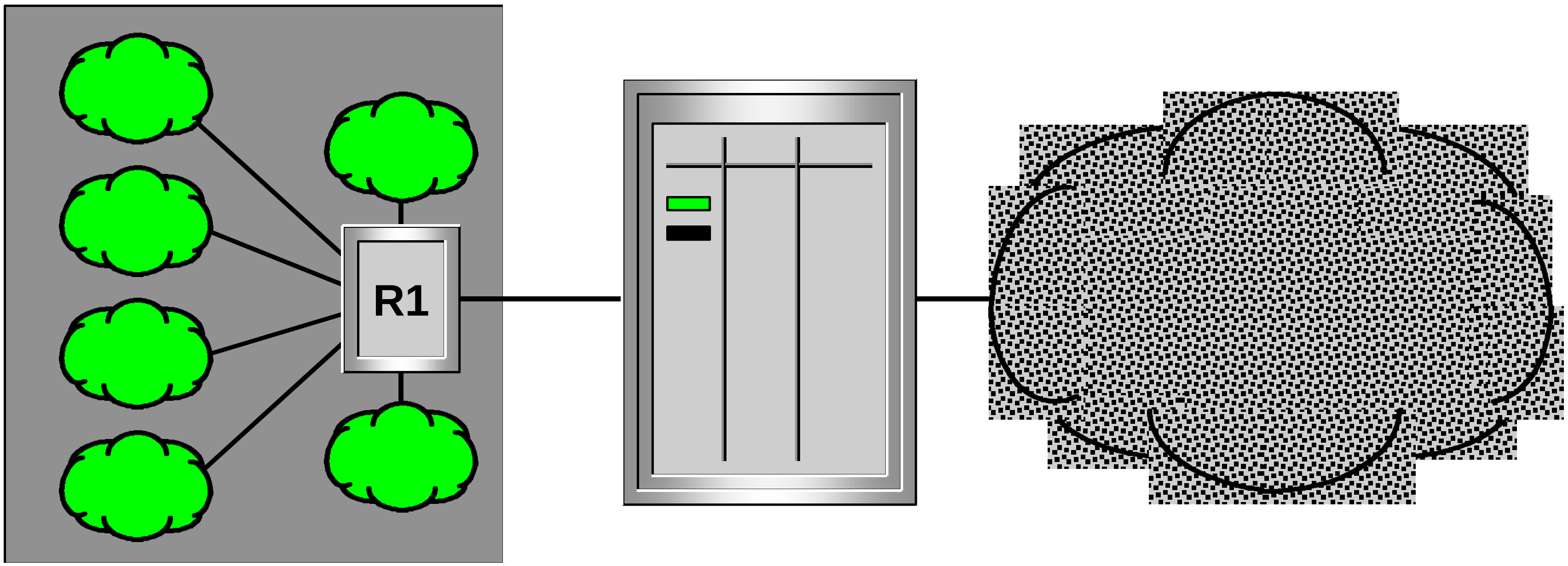
The situation becomes even more serious with an Internet attachment.

IPV4: Routing Table Size



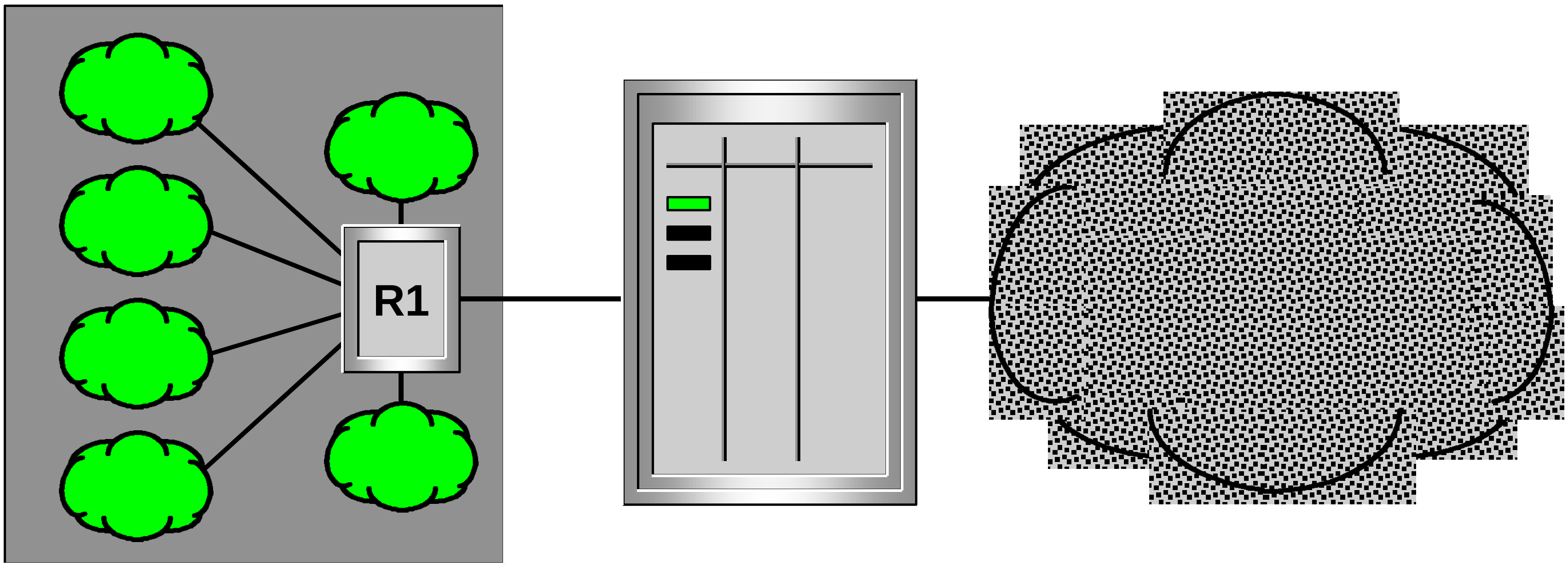
Imagine that each of these dots represents an IP network.

IPV4: Routing Table Size



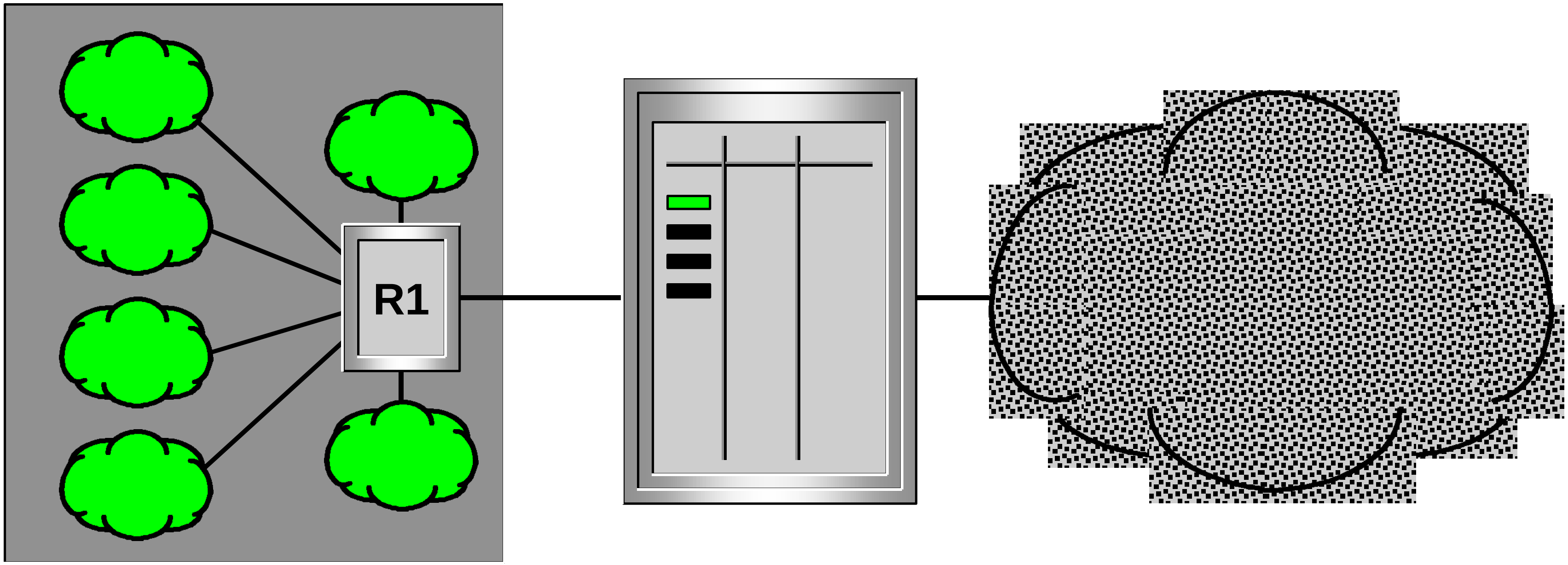
Imagine that each of these dots represents an IP network.

IPV4: Routing Table Size



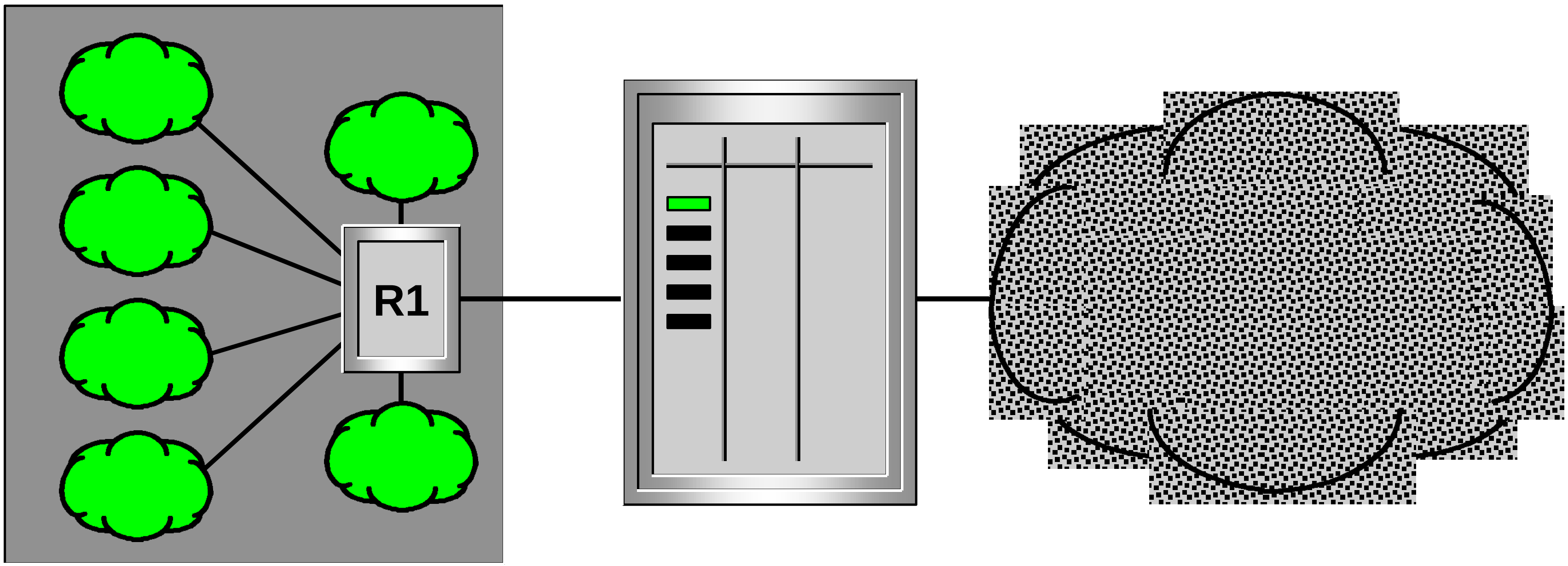
Imagine that each of these dots represents an IP network.

IPV4: Routing Table Size



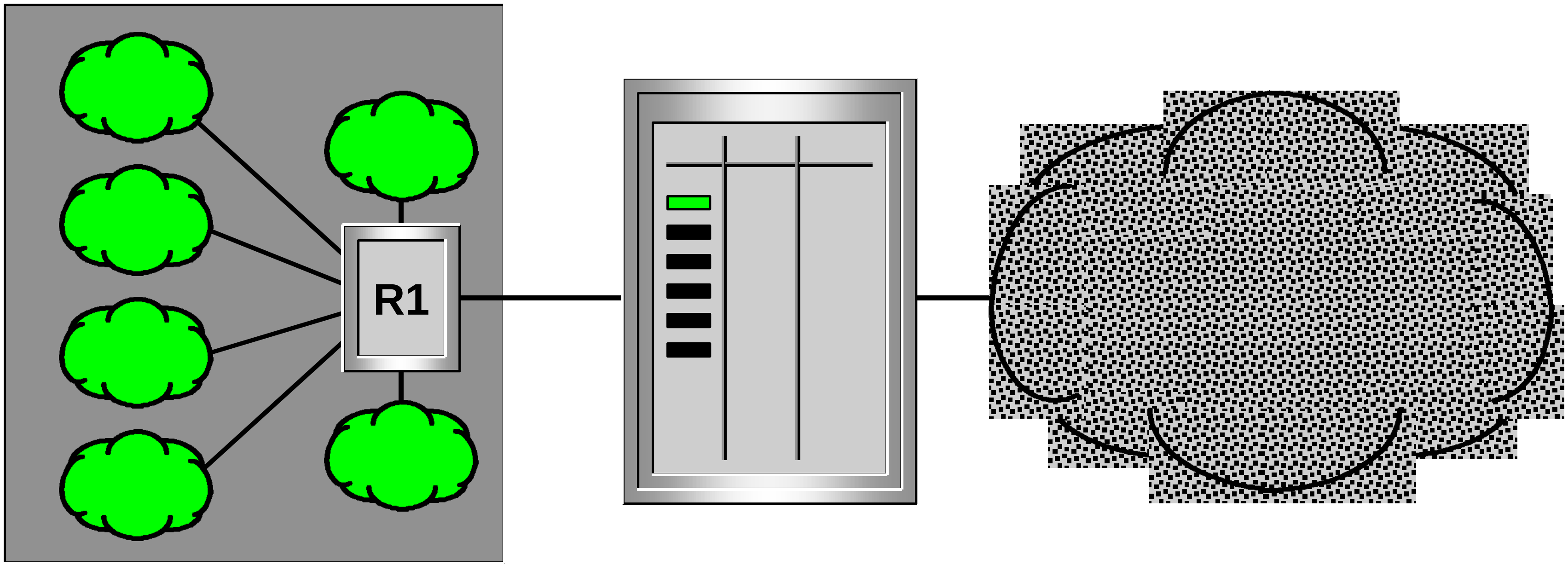
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.

IPV4: Routing Table Size



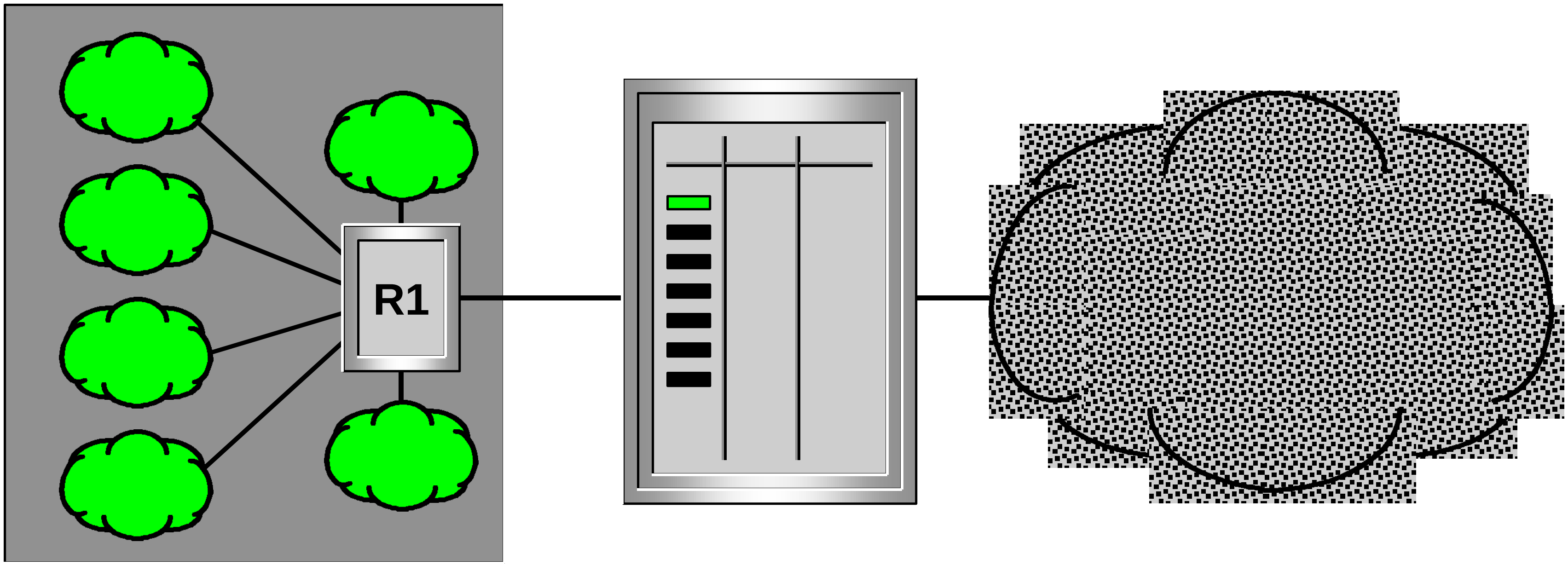
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.

IPV4: Routing Table Size



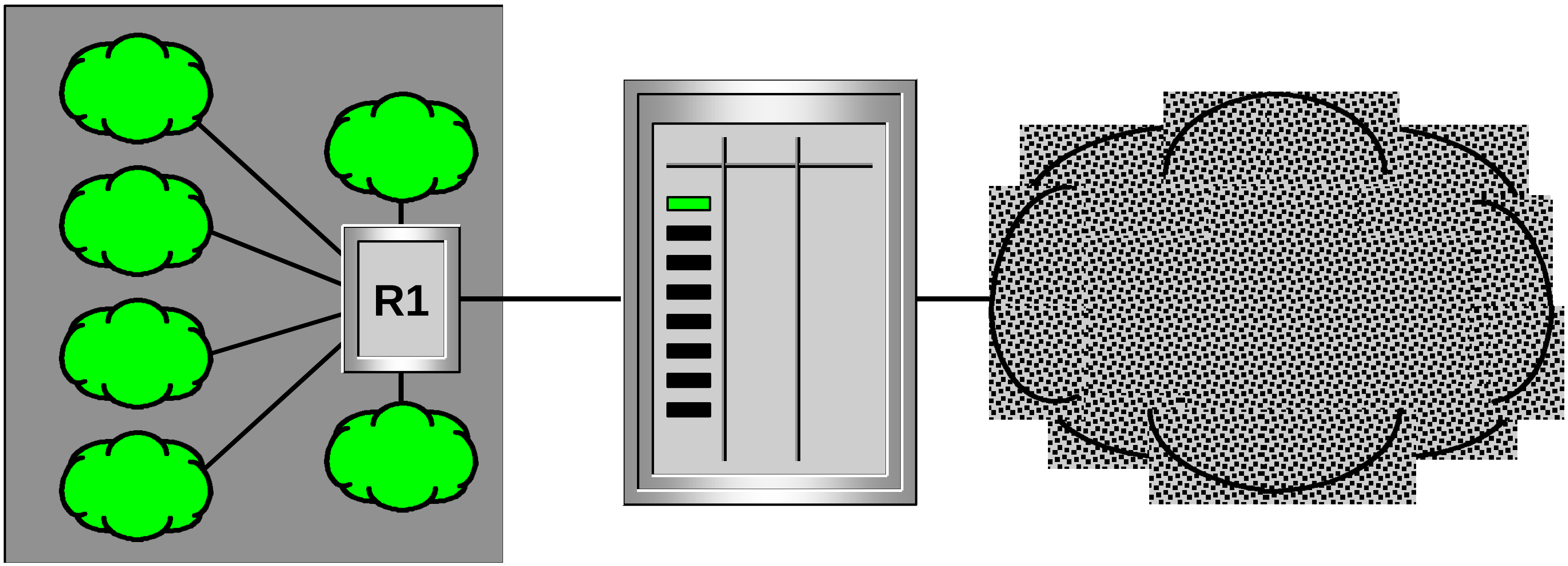
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.

IPV4: Routing Table Size



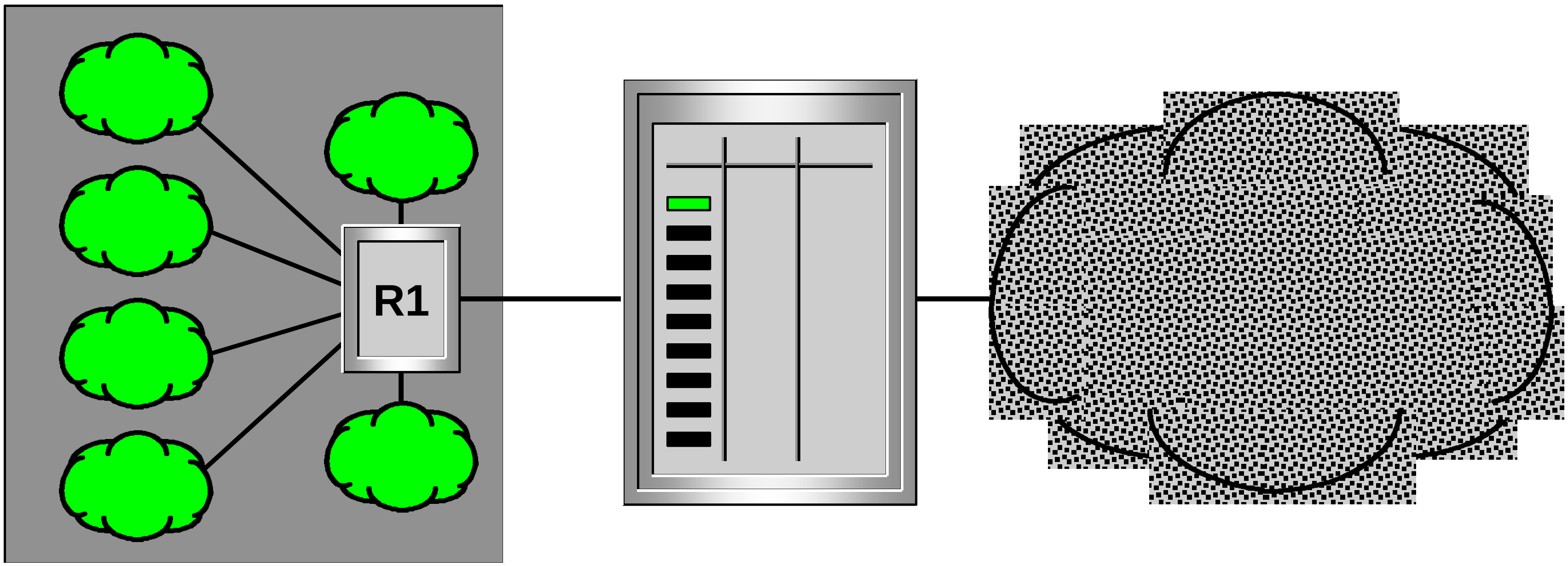
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.

IPV4: Routing Table Size



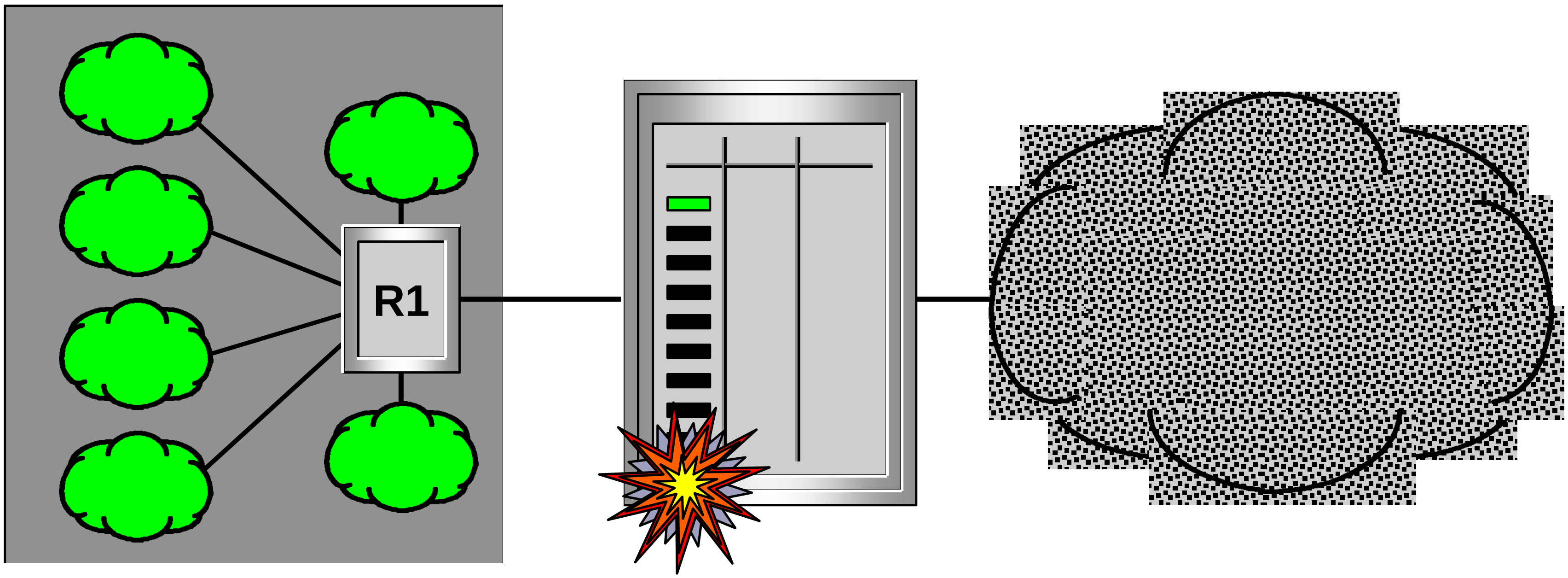
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.
And when the Table is full...

IPV4: Routing Table Size



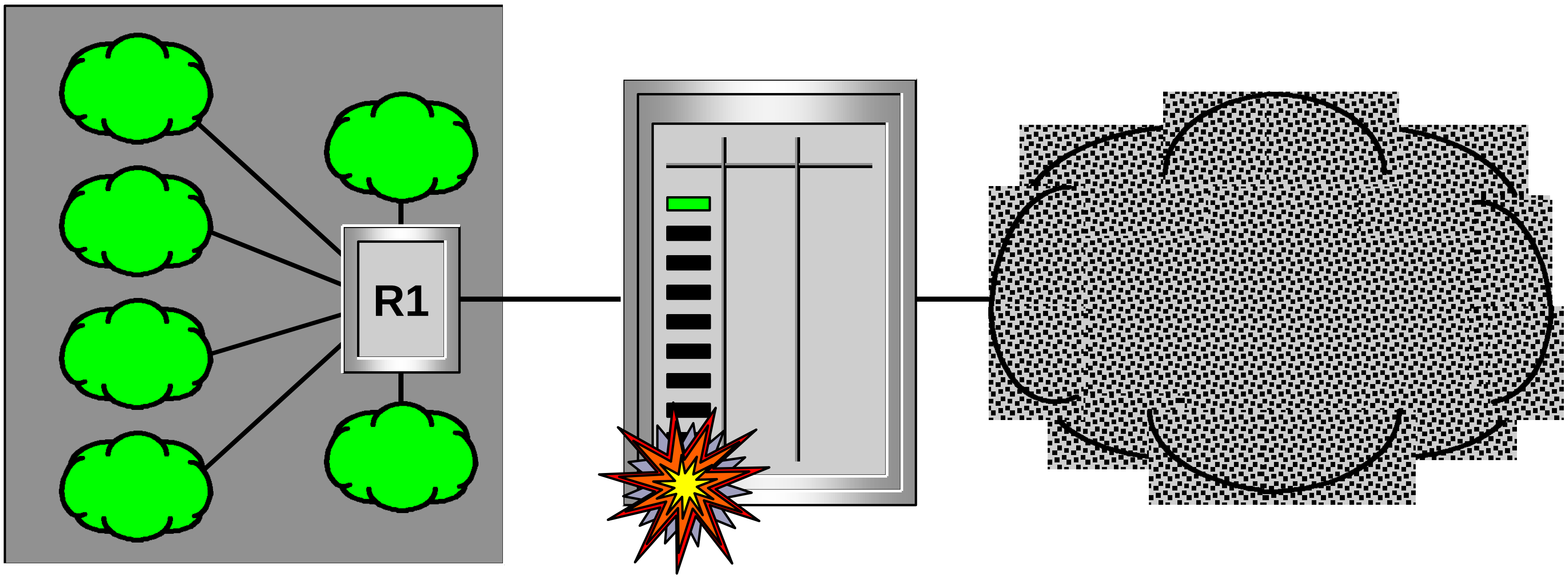
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.
And when the Table is full...

IPV4: Routing Table Size



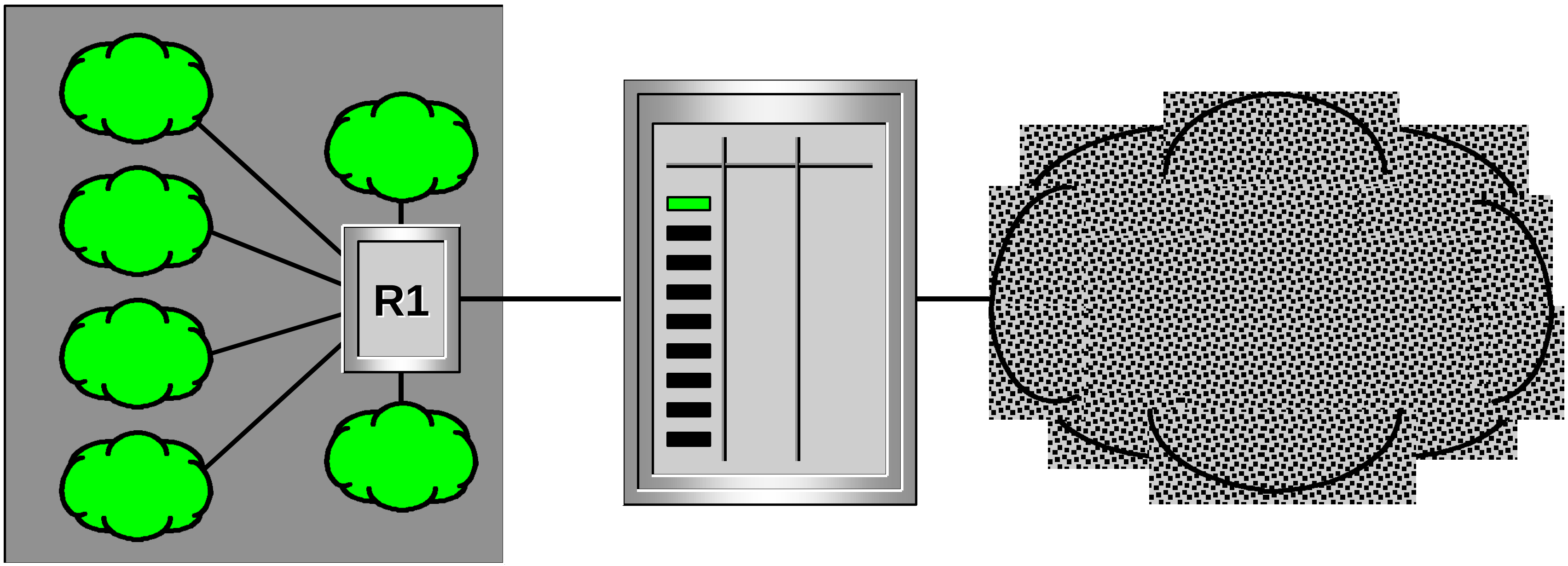
Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.
And when the Table is full...
...anything could happen!

IPV4: Routing Table Size



Imagine that each of these dots represents an IP network.
The Routing Table will fill up with Internet routes.
And when the Table is full...
...anything could happen!

IPV4: Routing Table Size

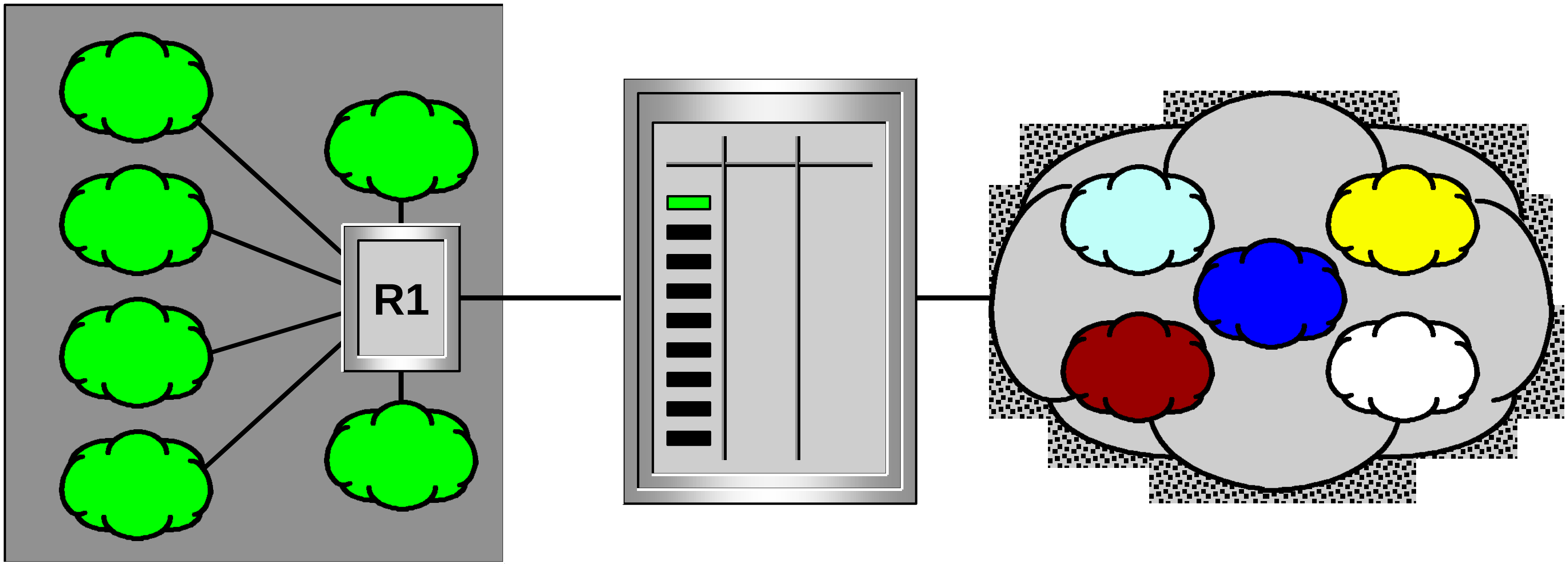


Ideally we should be able to combine routing information from each of these networks in a similar way to conventional IP subnetting.

Until recently this has not been possible because any address agglomeration would have to operate with current IP addressing and routing protocols.

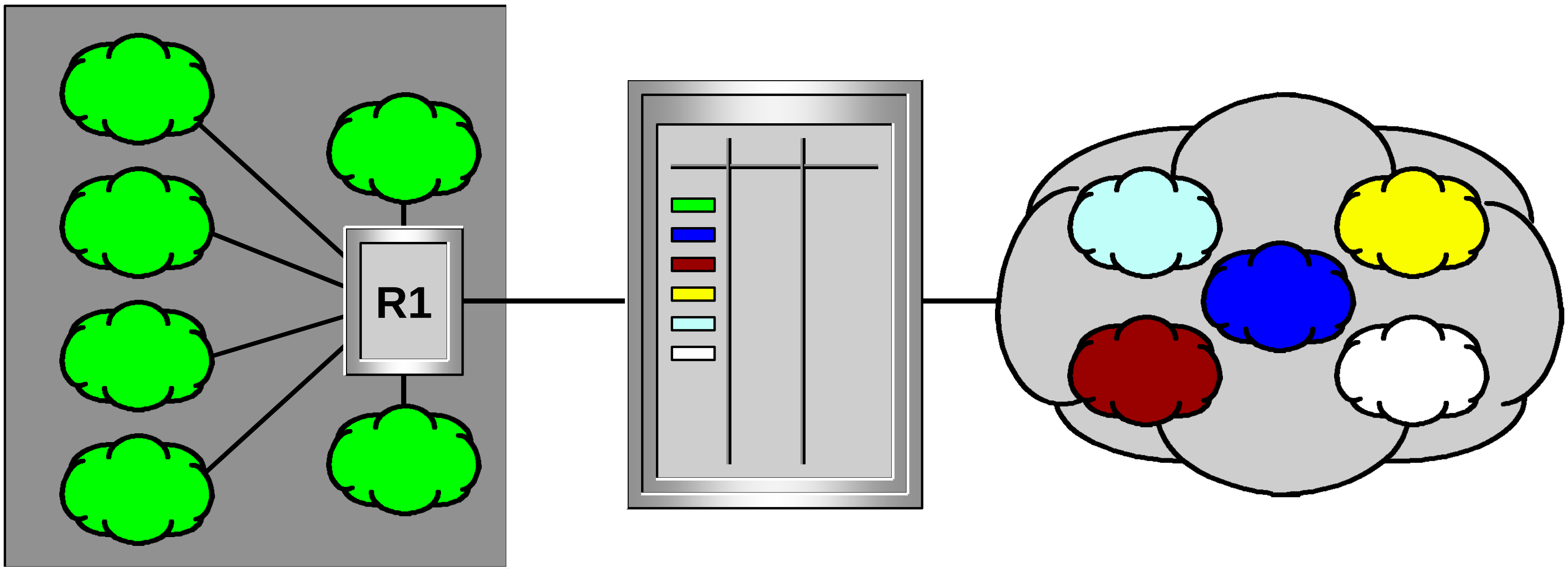
A new mechanism was suggested for routers which provide connection into the Internet, and this is known as Classless InterDomain Routing (CIDR).

IPV4: Routing Table Size



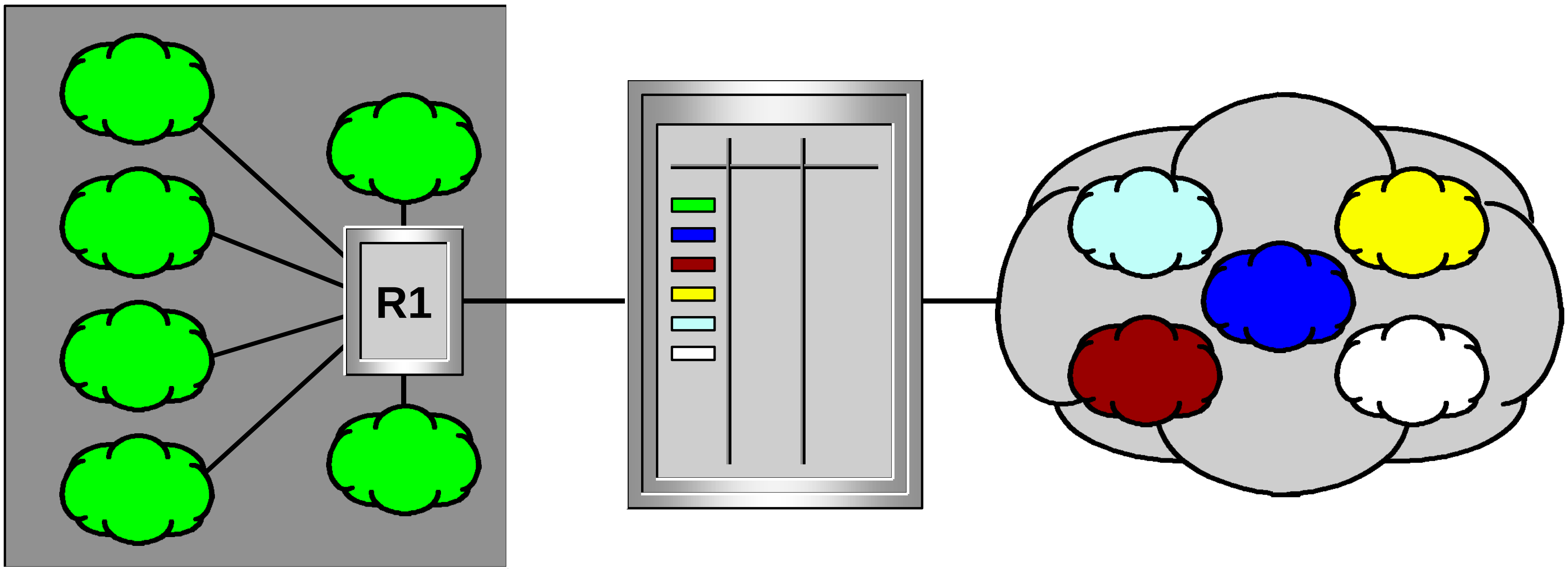
CIDR can make parts of the Internet appear as logical clusters of addresses.

IPV4: Routing Table Size



CIDR can make parts of the Internet appear as logical clusters of addresses.
And the Routing Table has a single entry for each logical cluster.

IPV4: Routing Table Size



CIDR has bought the Internet community time in which to develop a new generation of IP.

IPv6: The Next Generation

To solve the problems with IPv4, and to offer new possibilities for the evolving Internet, IPv6 has been developed.

Thanks to the CIDR initiative, IPv6 has been developed with extensive consultation, and has included consideration of a number of different proposals.

IPV6: History

- **IPAE**
- **SIP**
- **PIP**
- **TUBA**
- **TP/IX**

The initial debate over the next generation of IP, “IPng,” began by considering five different protocol proposals.

IPV6: History

IPAE

SIP

PIP

TUBA

TP/IX

IP Address Encapsulation (IPAE) basically proposed that existing IP networks be interconnected over an encapsulation scheme.

The most basic version of the proposal (“IP-in-IP”) simply encapsulated an IPv4 packet inside another IPv4 packet.

IPV6: History

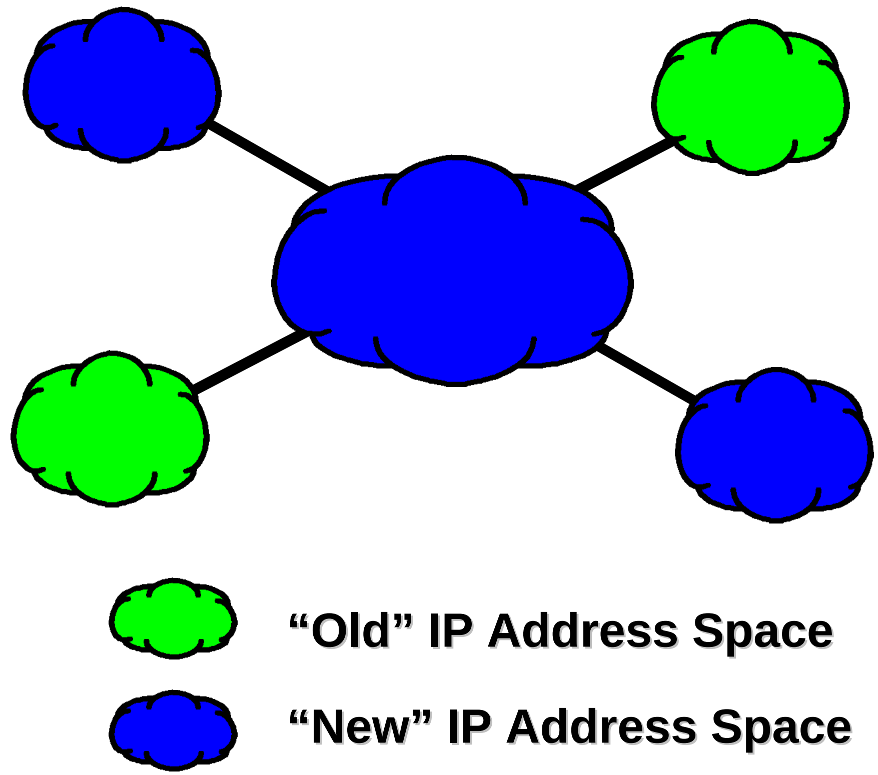
IPAE

SIP

PIP

TUBA

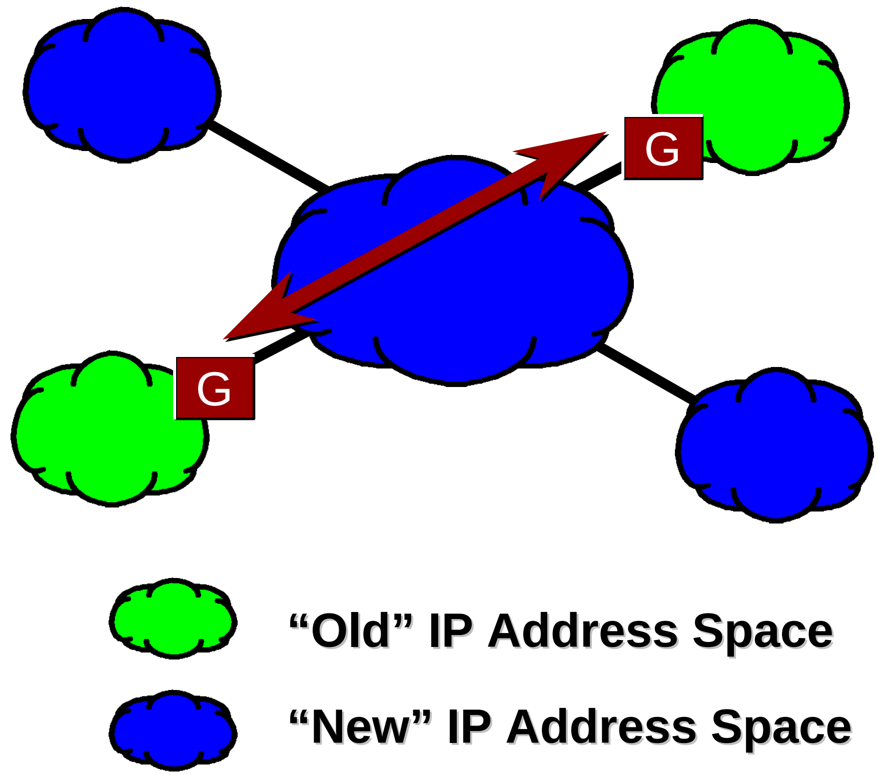
TP/IX



In this example we can see two IPv4 clouds are interconnected over a new IPAE core.

IPV6: History

- **IPAE**
- SIP
- PIP
- TUBA
- TP/IX



Encapsulation Gateways (G) are used to “wrap” the IPv4 in another layer of addressing, and to “unwrap” encapsulated IPv4 at the destination cloud.

Inside the IPv4 cloud, routing is done in the usual way.

IPV6: History

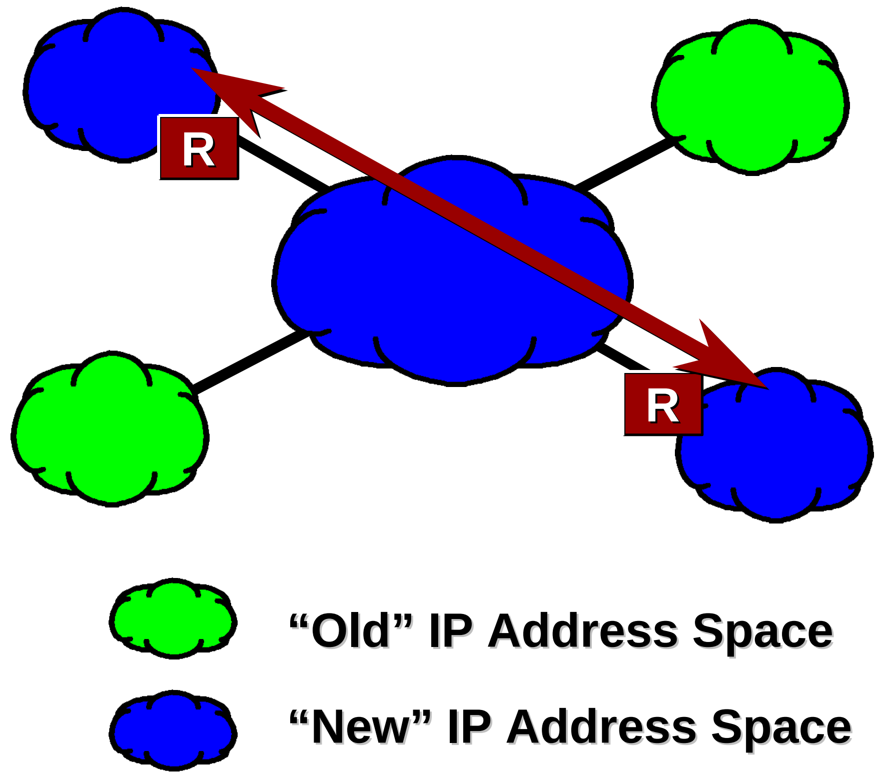
IPAE

SIP

PIP

TUBA

TP/IX



Connections between "new" IP devices are made using conventional routers (R).

IPV6: History

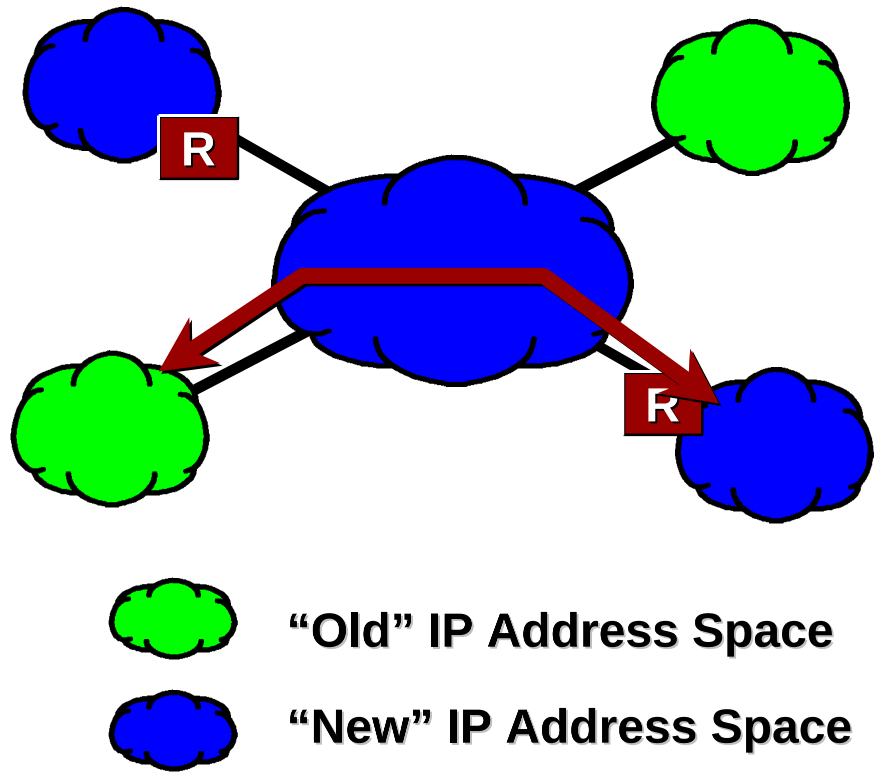
IPAE

SIP

PIP

TUBA

TP/IX



However, issues arise with this pathway, between the "old" and "new" address space.

In general, the IETF concluded that IPAE had too many problems, and that it would never scale to the level that was imagined for the future Internet.

IPV6: History

IPAE

SIP

PIP

TUBA

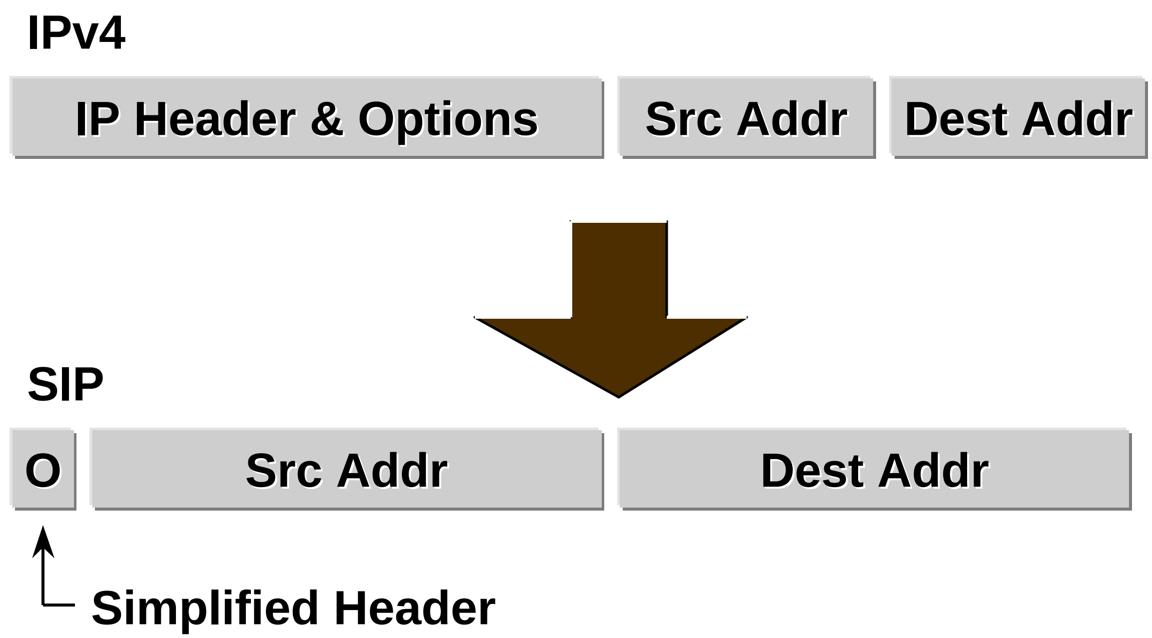
TP/IX

Simple IP (SIP,) and “Pauls” IP (PIP) were proposals that tried to simplify the IP header options to allow the address fields to be extended.

FORE SYSTEMS

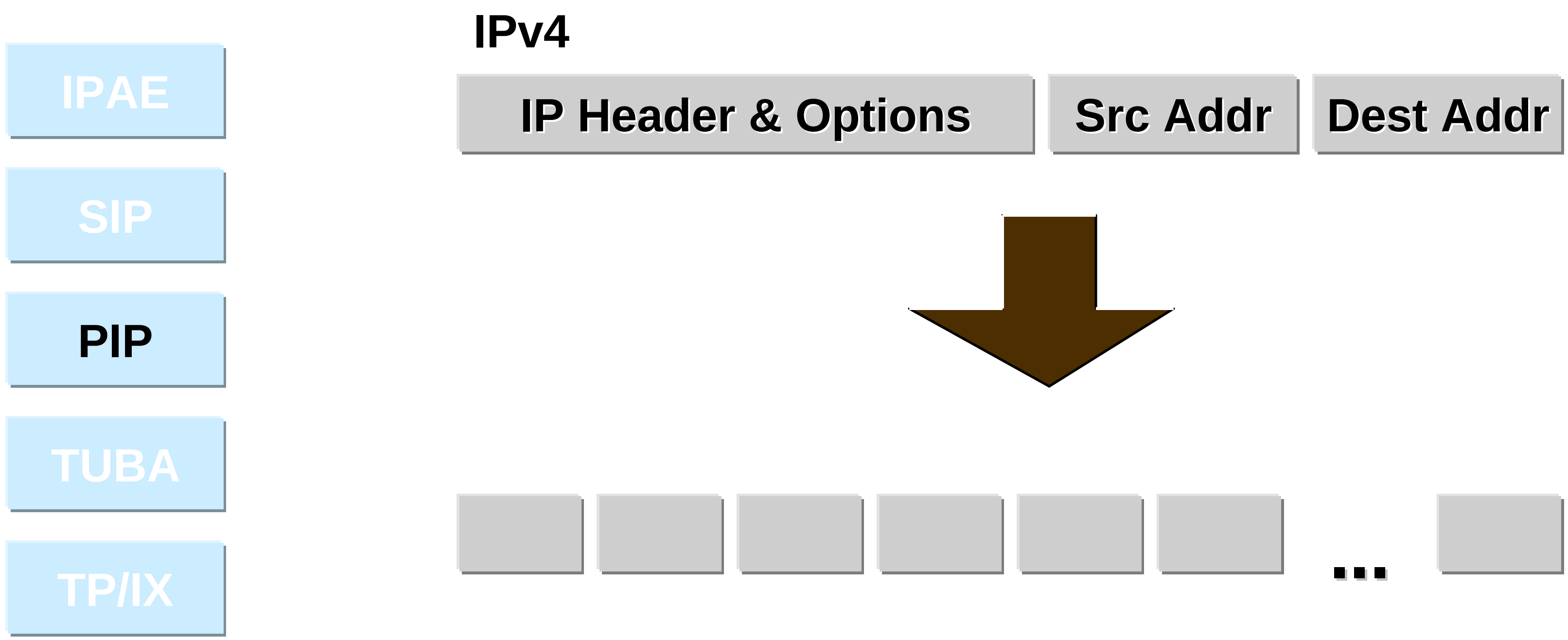
IPV6: History

- IPAE
- **SIP**
- PIP
- TUBA
- TP/IX



SIP began by simplifying the header of the IP packet, and using the space to extend the IP address to 64 bits.

IPV6: History

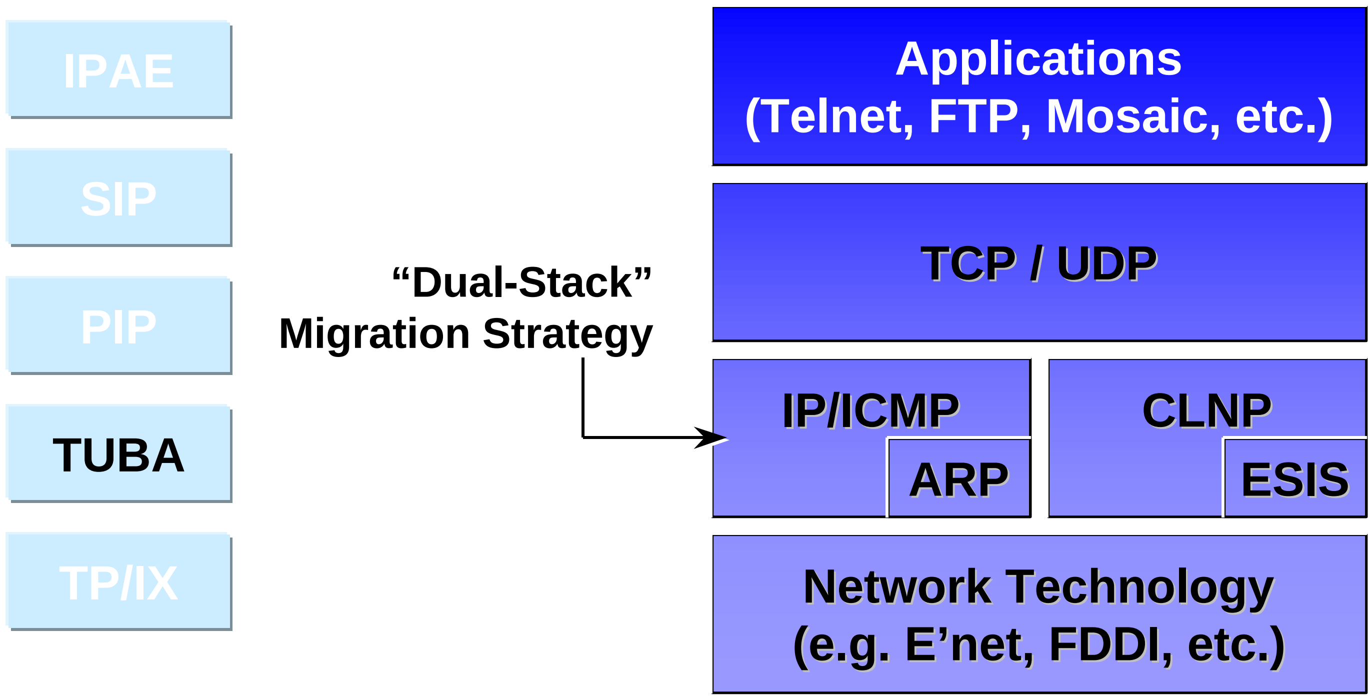


PIP included several radical proposals that would certainly make Internet growth more flexible, although it would make a router’s job a lot tougher in parsing the addresses.

PIP addresses were designed to be variable in length, in 16-bit blocks.

Special symbols were used in the address to select extensions, and to provide a number of other useful features.

IPV6: History



TCP and UDP over Big Addresses (TUBA) was the initiative taken by OSI supporters within the IETF. The "Big Addresses" were ISO Network Service Access Point (NSAP) addresses. These are a minimum of 20 bytes long, and can be extended.

TUBA was rejected for one simple reason (plus a few more complex ones). The IETF can't define changes that are necessary in the NSAP standard. This must be done by ISO. It was not clear that ISO would make the changes required by the IETF.

IPV6: History

IPAE

SIP

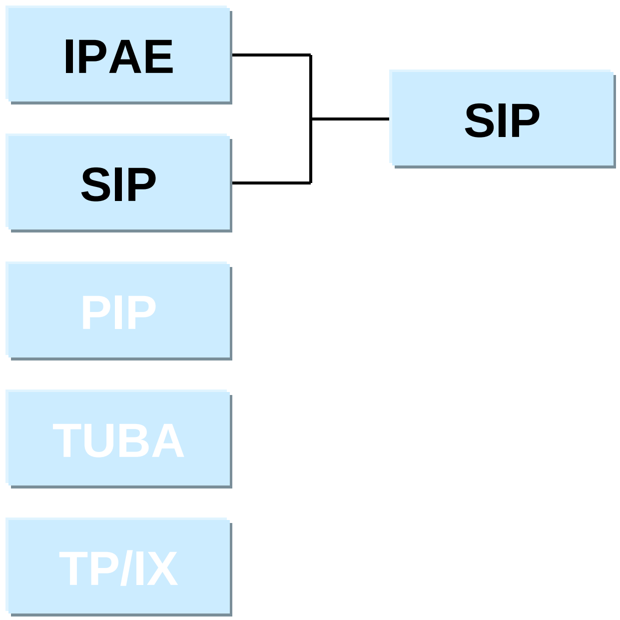
PIP

TUBA

TP/IX

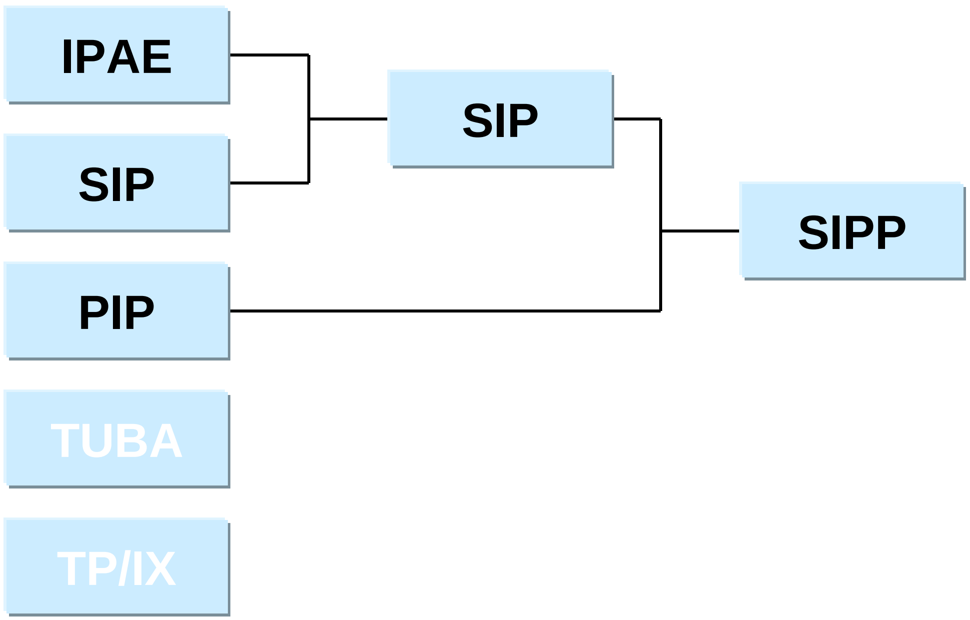
TP/IX, described in RFC 1475, began as an address extension for IPv4, and included the concept of 64-bit addressing, like SIP. In addition, TP/IX updates the TCP protocol to make best use of the new Network Layer.

IPV6: History



After consultation between the groups, it became clear that IPAE and SIP shared some fundamental characteristics. In particular, the transition mechanisms from IPv4 to IPEA could be used by SIP. So the groups supporting the individual proposals decided to merge their efforts.

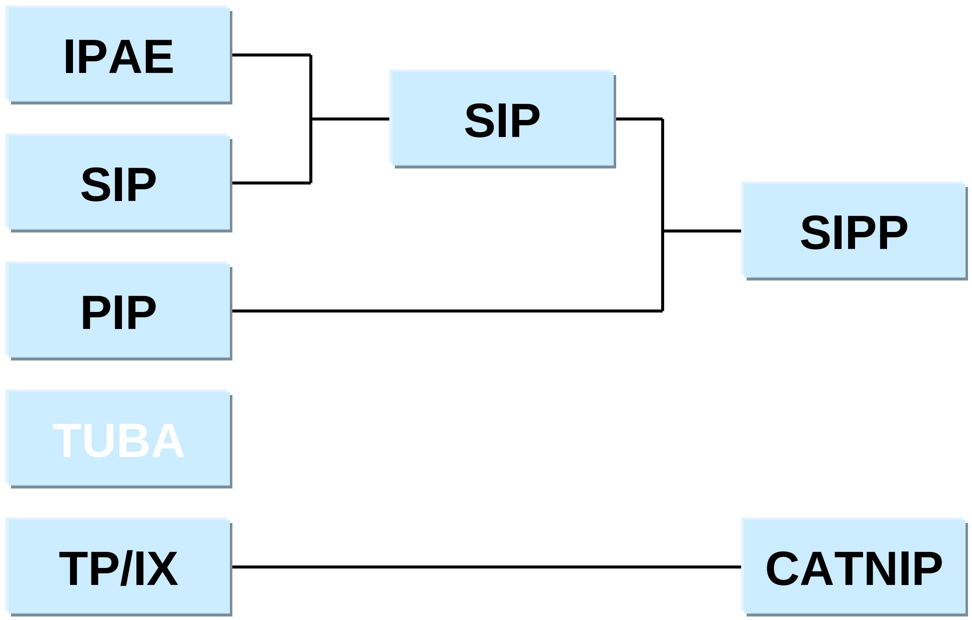
IPV6: History



After similar discussions between the merged SIP group and the PIP group, it was decided that PIP's advanced protocol features could be easily mapped onto the SIP architecture without losing the IPAE transition features.

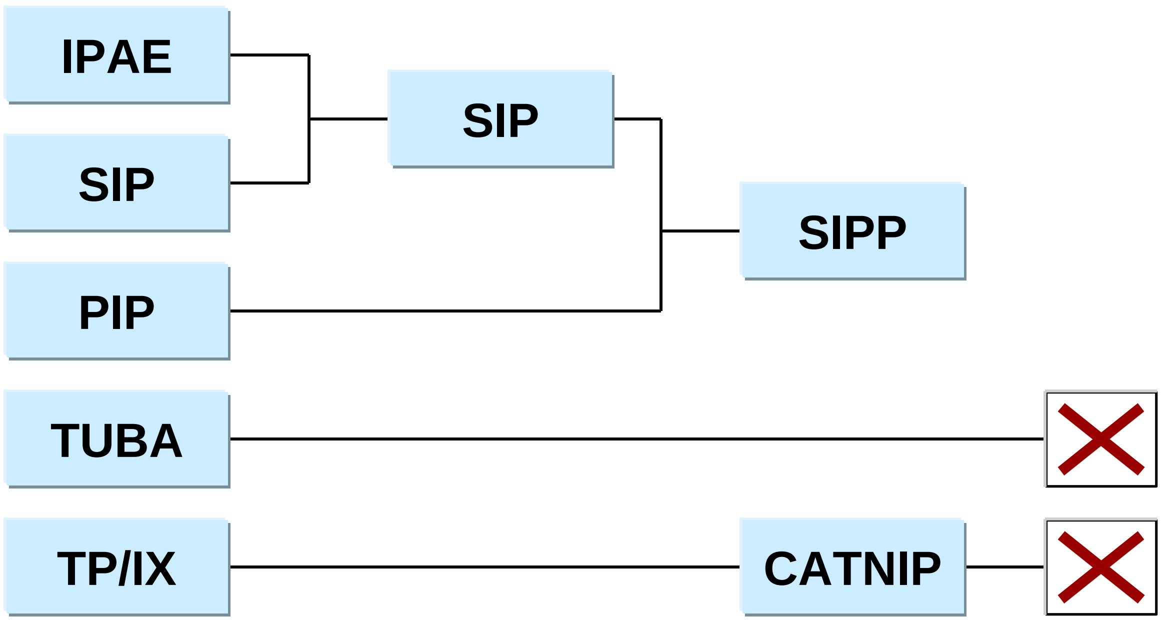
The combined group proposal is known as "SIP Plus" (SIPP White Paper, RFC 1710).

IPV6: History



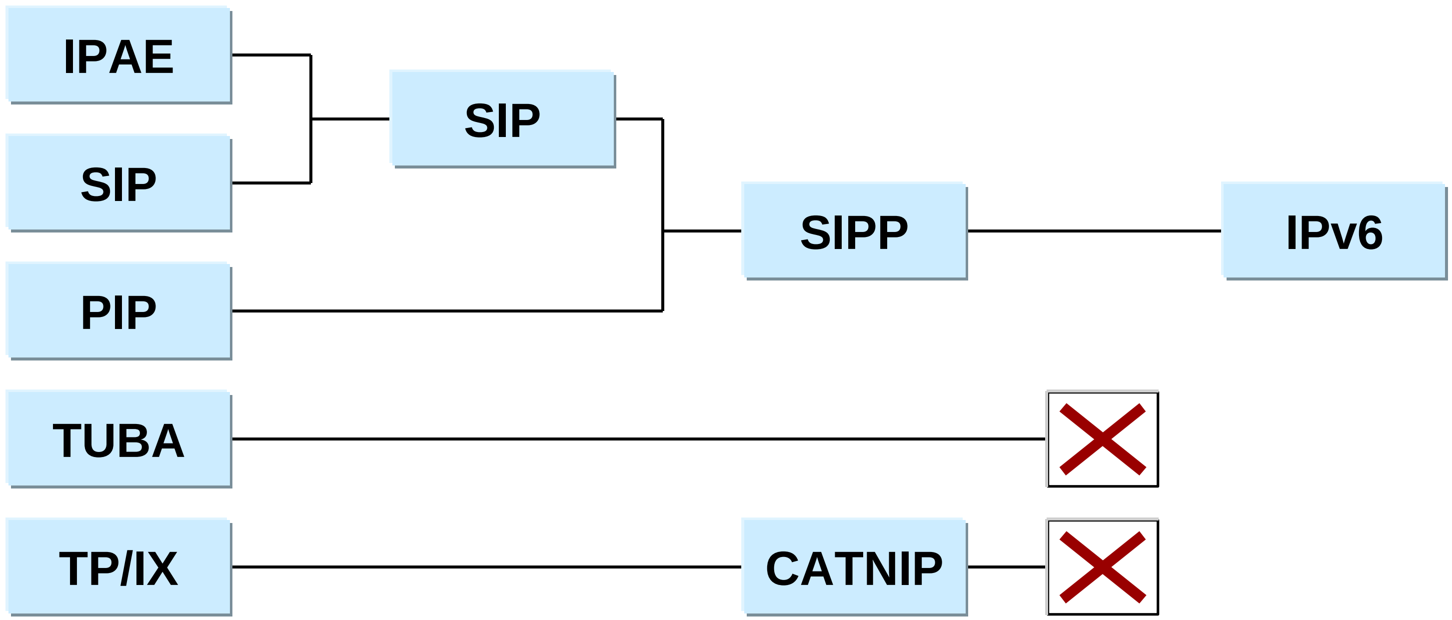
The TP/IX group continued its development, and included migration plans for Novell IPX. The combined proposal became known as the Common Architecture for Next Generation Internet Protocol (CATNIP).

IPV6: History



In the end, TUBA and CATNIP were rejected as candidates for IPv6.

IPV6: History



Instead, the SIPP proposal has been adopted, with extended 128-bit addressing.

FORE SYSTEMS

The End

This concludes the tutorial.

If you aren't viewing this tutorial on the FORE Systems' ATM Academy Site, then you can find additional tutorials at:

http://academy.fore.com/