lec 1 Introduction **High Speed Networks:** Provide high-speed information exchange
Quick Look at **Access** Networks ▪20 years ago: Dial-up Modem ▪15 years ago: ISDN ▪10 years ago: ADSL, CableModem ▪Now: Fiber
DSL—Digital Subscriber Line **Link Bandwidth**
Twisted-wire remains the same; Modem is different
DSL Bandwidth in Detail 图

VDSL (Very high bit-rate DSL) 图✎single channel---easy to control--anti-noise; has high utilization
1st Step to High Speed: Link
♦Advanced technologies: Broaden the road -->bandwidth
♦Cost-control + massive production: Affordable equipments
2nd Step: High-Performance Nodes ***Terminals:*** Directly affect the application performance ***Network nodes:*** Determine the performance of data exchange
3rd Step: Powerful Control and Management
Fixed network resource: Maximize the utilization IP Routing
Fixed performance criteria: Minimize the investment and maintenance Ethernet Spanning Tree ***Fixed network architecture:*** ▸Accelerate service provisioning ▸Fast failure restoration ***Stability of the network***
High Speed: Foundations High Speed Networks
▸**Link:** Mainly Hardware
▸**Node:** Hardware + Software
▸**Control & Management:** Mainly Software
Application ▸Interconnection of different types of networks ▸Hierarchical architecture
Links of Backbone Networks
Time-domain multiplexing (TDM)
▸Signals are interleaved in the time axis ▸Periodic slots turns to be a channel with *fixed* bandwidth ▸A link can be divided into multiple channels
Wavelength-domain multiplexing (WDM)
▸A single **fiber** contains several wavelengths
▸Each wavelength are relatively independent
Switching~
Backbone Networks
▸A fiber has a huge capacity, e.g., 320 Gbps ▸Wavelengths have coarse granularity, e.g., 10Gbps per wavelength
▸Applications need fine granularity, e.g. 1 Gbps
Question: How to handle the **mismatch**?
High Speed Networks
Physical Topology Fibers ✦OXC (optical crossconnect)
Logical Topology
Channels (slot channel or wavelength ✦Routers
Benefit of Logical Topology 图
▸Cost-effective ▸Better delay performance for cut-through logical links ▸Easy to manage packet streams
Switching Node --incoming links-(Node configured according to request**)--outgoing links**
Packet Switching --has memory
IP/WDM IP/SONET/WDM IP/SONET IP/Eth/SONET/WDM
Discussions: Is packet switching better than circuit switching? Can we use circuit switching to support packet switching?

lec 2 SONT/SDH --optical network
SONET: Synchronous Optical NETwork ANSI
SDH: Synchronous Digital Hierarchy ITU-T
Interoperability between SONET and SDH
SONET is **circuit switching**
C1...--(1G)--A==(10G)==B--(1G)-- C2...
between A and B--fiber inside multiple wavelength [TDM (time slot) over WDM]; custom C1 & C2 have socket, with fix bandwidth
Fiber Optics Look into a Fiber
Multimode ▸Thick core ▸Dispersion ▸LED/Laser
Singlemode ▸Thin core ▸Laser ▸Long distance
3R- at receiver we should regenerate the signal
①Reamplifying ②Reshaping ③Retiming---(physical layer)
E1: A Simple TDM 图
Voice channel--Sample frequency: 8 kHz; Encoding: 8 bits
why data synchronize?
sampling is done in one place/machine (base on one clock)
▸采样周期=125us ▸划分为 32 个 time slots ▸每个 slot 传送 8bit
▸总共用 8*32=256bit ▸每秒传送 8000 个 frame(8KHz)
▸传输速率=256*8000=2.048ubps
2 time slot not used:
TS0---frame synchronization 1010110 (what is the boundary) device know the R-G system ready to use
TS16--signaling--control network--which time slot is used to connection? other TS-payload---carry user data
E1 frame 图(u1 u2 u3)

Frame Synchronous Scrambling random--avoid payload repeat(10101100) like frame header
R: random number **XOR**: exclusive OR(无进位的二进制加)
=>
Receiver:1st: synchronize--2nd: use *same* R to XOR
SONET Frame 图 **speed** and **bandwidth** is more than E1
32 user -each user has 1 byte=>32 bytes

E1--> SONET frame **9 rows*90 bytes(column)**
(send row by row) still occupy only one time slot

the duration of frame always=125us(fixed)---speed faster(51.84Mbps) 传的数据多 ; frame rate always 8KHz
first three column are **overhead**--synchronization and other signaling use ▸first 2 bytes(maybe)--do not scramble--just skip the part of synchronization
Rate Mismatch?
Circuit A: 1 Mb/s Circuit B: 1 Mb/s
Mux A & B to C:(A&B running their own clock)
Ideal: 2 Mb/s is enough for the payload
But: the clocks of A & B are not precisely the same
So: capacity of C is **slightly larger than** 2 Mb/s
Bit stuffing --why little more than? 图
(a,b,c): two 0s, x is useless, no stuffing
(a,b,c): two 1s, x carries information, stuffing
Hierarchical Multiplexing--low speed->high speed
Ideas of SONET
▸The whole network is synchronized ▸Standard frame format for hierarchical multiplexing ▸Embedded overhead channels-- a lot of overhead can be use ▸Survivable rings failure-recovery for itself automatically
Clock & Data--synchronization
Data read/write is based on clock--without clock, how many "1"s? don't know Send a separate clk for long distance transmission is not feasible, why?---delay could be different ; So inside the data, we insert some clock information--can recover clock
Clock Data Recovery **PPL**(phase locking loop)--recover local clock same as it send--very critical
signal --compare with local clock --slightly faster/slower--**adjust** local clock ▸Local oscillator ▸External reference ▸Adjust ***local clock*** according to the reference
Synchronization of SONET
preamble-10101011(give enough information for synchronize) **Tree**
node A(stratum 1)--(time signal)--->node B(stratum 2)
▸Hierarchical network synchronization
▸Stratum 1: atom clock with extremely high stability and accuracy ▸Less stable clocks are adequate to support the lower nodes
Retiming for Synchronization
F*=not accurate clock
F=standard/accurate clock(master give?)
Master--Slave
retiming: use F* to receive --[buffer]--use F to send
SONET Layers **terminal=mux-reg-reg-reg-mux=terminal**
▸***path***--between terminal ▸***line***--between Mux
▸***section***--between every hop(reg-regenerator)
▸physical layer--optical
POH: Path OverHead **SPE**: Synchronous Payload Envelope
LOH: Line OverHead **SOH**: Section OverHead
STS-1 Frame
[(9 rows(其中 SOH 3 rows; LOH 6 rows)+9rows)*90bytes]
STS-1 Synchronous Payload Envelope (SPE)
图

not precise aligned--▸wait? need buffer ▸do not wait (continuous transmission)- just put in--but where to start???-
Pointer E1 - no pointer; but SONET frame has pointer
A pointer indicates the offset from the starting point of a SONET frame to the starting point of the payload frame, thus can be used for positioning of the payload. Using of pointers can reduce the buffer requirement as well as multiplexing delay.
STS-N Frame [(9 rows+9rows)*90*N bytes(其中 transport header=3*N bytes)]
SONET MUX with Frames Aligned--always 125us
SONET MUX with Frames **Un**aligned▸STS–3 sends out frame at arbitrary time ▸Use **pointer** (H1 and H2 bytes) to point to the 1st bytes of the payload ▸3 tributaries needs 3 pointers ▸Save the memory at each input
SONET Add-Drop Multiplex **(ADM)**
(a)pre-SONET mux:
DeMux (remove tributary)---Mux(insert tributary)
(b)SONET ADM:
use ADM instead a demux+mux do remove /insert tributary (time slot subway)
OAM-P Operations Administration Maintenance Provisioning
Networking ▸Traditional networks are **point to point**
▸SONET is a **unified** system
SONET Rings in Metro Networks
Ring has two advantage:
(1)simple (2)resilience- has two directions
Automatic Protection Switching (APS)
(a) Dual ring (b) Loop-around in response to fault
Each link contains a working channel and a protection one along different directions, which is called **1+1 protection**.
Upon a link failure, the end nodes detect the abnormality, send out notifications and switch the working channel to the protection one by pig-back. APS takes a short time (usually 40ms) to avoid/reduce service interruption.
Lec 3 Asynchronous Transfer Mode (ATM) and Multi-Protocol Label Switching (MPLS)
Basics ▸ATM and MPLS are packet-switching / statistical multiplexing ▸ATM and MPLS support connection-based communications ▸Background: Integrated Service
Requirements ▸Integrate multiple services ▸Support **fast** packet forwarding--high capacity/more bandwidth ▸Provide various quality of service guarantee
QoS: bandwidth, delay, delay jitter, cell loss rate

ATM Basic Concepts
Negotiated Service Connection End-to-end connections, called virtual circuits Traffic contract
Switch Based Dedicated capacity
Cell Based **Small**: the requirement of buffer size is small
header + payload --> grasp header->table look up->forwarding
if long length->need more time to processing /queuing time long
fixed length: hardware design is easier
Negotiated Service Connection
IP-best effort--network does not provide any guarantees that data is delivered or that a user is given a guaranteed quality of service level or a certain priority
The ATM **Cell** 图
Small Size: 5 Byte Header + 48 Byte Payload(53)
Fixed Size ▸Header contains virtual circuit information
▸Payload can be voice, video or other data types
Cell Header What kind of overhead do we need?--
Depends on the operations
Where do you go?--*Address/identifier*
IP- destination IP add; TCP-destination port
What do you carry?--*Type*
How important?--*Priority*
Ensure there is no error!--*Error Control*
Cell Header Details *[5 byte]*
♠VPI: virtual **path** identifier(1.5byte=20bits)
♠VCI: virtual **channe**l identifier (2byte=16bits)
♠-tell you which point to which point?
PTI: payload type identifier
CLP: cell loss priority
HEC: header error control detection
Virtual Paths (big pipe)and Virtual Channels(small connection)
▸Bundles of (VCs) are switched **via** (VPs) ▸Virtual Path service from a carrier allows ▸reconfiguration of Virtual Channels without service orders to carrier ▸VP switching does**n't** change VCI ▸VC switching **changes both** VPI and VCI
assign label if consider both VPI&VCI(only consider **one**) table size will very large
Discussion
Advantage of VP+VC switching?
Differences between ATM switching and IP routing?
ATM： *connection already*label based(focus on connection)
IP： *not connection already*destination based(destination)
Network Interface
UNI: User network interface
VPI(first 4 bits --generic flow control[GFC])
NNI: Network-network interface
Payload Type Identifier (PTI) what kind of payload
Cell Loss Priority Bit ▸Cells with CLP = 1 will be discarded before those with CLP = 0 ▸Can be set by the terminal (e.g., video coding). ▸Can be set by ATM switches for internal network control: ▸Virtual channels/paths with low quality of service ▸Cells that violate traffic management contract
Header Error Control
Detection mode Discards cell when header error
Correction mode (optional): Correction 1 bit errors else discard when error detected
Reduced cell loss for the case of single bit errors
Cell delineation--can be done by use HEC header. Find the start & end, identify the boundary of ATM cell
Question: Consider a cell spanning multiple hops, does the HEC change or keep fixed? yes, it does change! because an ATM cell is delivered across a network **The HEC may change. The field is calculated based on the other fields of the header. Since the VCI, VPI, etc may be changed at each intermediate node, the HEC needs to be recalculated accordingly.**
Ethernet frame： CRC do no change.
Through Router--- the destination MAC address is changed, CRC should recalculate. IP header also change--TTL change
Receiver HEC Bimodal Operation 图
Detection Mode -- Correction Mode
just correct single bit error
Cell Delineation State Diagram
what is the start of a frame.
grasp one [4 byte]-->calculate the HEC-->compare with the next one byte --(1)correct--- come later 53 bytes=>find next header (2)coincidence
HUNT--PRE-SYNC-SYNC 图
packet delineation. each IP packet has a 20-byte header, in which two fields can be used for delineation: Length and Checksum. At the beginning, we can search byte-by-byte or bit-by-bit for an IP header. A header is confirmed if the checksum from our calculation equals to the assumed checksum received from the bit stream. After the previous step, we start a counter to count the number of bytes from the IP header. Based on the Length of the current packet, we can easily locate the beginning of the next IP packet, which is verified by calculating the checksum again. The state machine and state transfer diagram is the same as that of ATM.
Service Categories
ATM was designed to provide multiple service support
Real-Time Services
Constant bit rate (CBR) ▸The simplest ▸Fixed bit rate, tight delay bound ▸Examples: uncompressed audio, video
Variable bit rate (rt-VBR)-flow bit rate change all the time
▸More flexible than CBR ▸Delay-sensitive, bandwidth variable
▸Better multiplexing gain ▸Examples: compressed media stream
Non-Real-Time Services **nr-VBR*** Provide peak and average bit rate *No delay guarantee **Unspecified Bit Rate (UBR)** *Best-effort *To utilize the unused bandwidth
Available Bit Rate (ABR) *Minimum bit rate guarantee*With feedback **Guaranteed Frame Rate (GFR)** *Aware of frame/packet boundary *Improve goodput rather than throughput
ATM System Architecture
▸ **ATM Adaptation Layer**
-conversion to ATM data types, 48-byte length
Provides Mapping Of Applications To ATM Service Of The Same Type; Segments/Reassembles Into 48 Payloads
Hands 48 Byte Payloads To ATM Layer
▸ **ATM Layer**-forward cell through network; add 5-byte header Adds/Removes Header To 48 Byte Payload
Header Contains Connection Identifier; Multiplexes 53 Byte Cells Into Virtual Connections; Sequential Delivery Within A Virtual Connection ▸**Physical Layer**-convert to correct electrical or optical format

MPLS Terminology
LDP: Label Distribution Protocol LSP: Label Switched Path
FEC: Forwarding Equivalence Class LSR: Label Switching Router
LER: Label Edge Router (Useful term not in standards)
MPLS: Big Picture *Multi-Protocol Label Switching
Attach a label* to each packet *Labels are generated according to addresses and QoS requirements *Labels are generated by **edge routers ***Core routers** perform packet forwarding according to **labels**
simply add MPLS header--still IP packet with layer 2 label
[Layer 2 header][MPLS header][IP header]
-- only look up MPLS label
MPLS do not has error detection code - supported by layer 2 - Ethernet take care about error detection
MPLS Label much simple than ATM
An MPLS header has 32 bits
Label (20 bits): used for switching **Exp (3 bits)**: usually used for QoS priority **BoS (1 bit)**: 1-bottom of stack; 0-not the bottom. (label stack will be explained shortly) **TTL (8 bits)**: time to live, same function as TTL in IP header
MPLS Advantages
▸**Short label** ·fast forwarding ·Hierarchical labels
▸**Scalable architecture** ·Edge routers do classification and labeling ·Core routers do forwarding ·Support multiple protocols ▸**Traffic engineering** ·QoS-aware routing ·Route adjustment according to network load
Features of MPLS Labels
·An MPLS label indicates a logical path, it does **not** represent any end host.----do not know the destination(same as ATM) ·physical location different ---path defined be MPLS ·An MPLS label has local meaning, it is used to distinguish logical paths on the same physical link. ·MPLS labels can be reused. ·MPLS label is **swapped** from one hop to the next. label will be changed
MPLS and TCP/IP ·Layer 2 formats the frames on a point-to-point link, performs error detection, and (optionally) provide flow and error control ·Layer 3 controls global address space, routing and network interconnection.
·MPLS does not fit into either layer 2 or layer 3. layer3- hop count function ·MPLS is often considered as layer **2.5-** not do any IP operation
Frame Structure ·The MPLS label appears **after** the layer 2 header ·The layer 2 header indicates the receiver at the link layer ·The MPLS header is used for routing (often times it is used to replace layer 3 routing) ·The payload of MPLS frame could be anything
Ex **Net 1--Ethernet switch A--Edge LSR B--Core LSR C--Edge LSR D--Ethernet switch E--Net 2**
Suppose all the interfaces are Ethernet, and a packet is sent from Net1 to Net2
A - B: [MAC header][IP datagram] ·The MAC header is created by a device in Net1, the destination is LSR B ·The IP header is created by a device in Net1, the destination IP address is in Net2.
B - C: [MAC header][MPLS header][IP datagram] ·The MPLS header is inserted by LSR B ·The MAC header is created by LSR B, the source/destination are B/C
C - D: [MAC header][MPLS label][IP datagram] ·The MPLS header is modified by LSR C using label swapping ·The MAC header is created by LSR C, the source/destination are C/D
D - E: [MAC header][IP datagram] ·The MPLS header is removed by LSR D ·The MAC header is created by LSR D, the destination is in Net2
IP FORWARDING USED BY HOP-BY-HOP CONTROL
MPLS Label Distribution Label in--label out
Label Switched Path (**LSP**)
compare link-based and path-based protection scheme for advantages and disadvantages. Failure recovery delay: Link-based protection is faster than path-based protection. There is extra signaling delay in path-based protection. In terms of resource complexity, path-based protection occupies less resource. Generally speaking, when a protection LSP is established, along that LSP each link needs to reserve bandwidth for protection. During normal operation, this bandwidth is not used. Therefore, we want to minimize the bandwidth used for protection. Compare these two designs, link-based protection requires more LSPs, the total number of link being used by such LSPs is larger, and the total bandwidth is more.
EXPLICITLY ROUTED LSP- the sender LSR can specify an explicit route for the LSP. Explicit route can be selected ahead of time or dynamically.
create multiple pipe-- give different label--go different way(path)===>more controllable than IP
MPLS Stacking two path share same section ,you can bundle them together---add another label(inner label & outer label)--just look up **outer label**
Label distribution Concept: notify LSRs which labels should be bound to a flow (e.g., an destination IP address) to form an LSP
Label distribution methods
Most widely used methods
Label distribution protocol (LDP): designed to bind labels to IGP (interior gateway protocol) prefixes

RSVP (resource reservation protocol): designed to provide MPLS traffic engineering

Other methods-Manual distribution: very flexible but incurs high complexity -Integrated with routing protocols: needs extension of the existing protocol, may have compatibility issues

LDP example Label distribution is from downstream to upstream. An LSR chooses is local label and creates a label swapping pair 交换对, then puts it in the forwarding table

Before LDP, the routers should have completed IP routing calculation (e.g. using OSPF)

Forwarding after LDP.. -Forwarding tables are constructed using LDP -Packet forwarding is now based on the assigned MPLS labels

Lec 4 Ethernet & Ethernet over SONET (EoS)

Ethernet: Quick Review

Basic Ethernet Implementation-Broadcast

everybody is synchronize to the source

Whoever transmits owns the wire!

CSMA/CD to control -If the bus is busy, do not transmit -If a collision occurs, back off and retransmit

-What is the problem of this network? not full duplex--A->B but B cannot ->A simultaneously -collision (not good scalability)

-How to improve the scalability?

1 Still use buses, but multiple domains

2 Use switches --divide to small domain (分开 collision domain)

Switched Ethernet

A switch has an internal **forwarding table**. When it received a packet, it performs the following: -Gets the destination address from the packet header -Performs table lookup using the address -Obtains the output port number -Forward the packet to that output port

If both send to C simultaneously? contention

queue-buffer --go to destination one by one, no collision

In switch base Ethernet-we do **not** need CSMA/CD.

Physical Layer: Go Faster...

wire has limited capacity~ how to improve speed?

10M→100M→1000M→10G

Motivations-10→100→1000, faster LAN

-10G, extend Ethernet to WAN

Challenges-How to accelerate the link speed (still using the twisted pairs) -How to maintain compatibility

Unshielded Twisted Pair(UTP) -Differential signals -Low interference -Low cost -Easy to deploy

How can we use UTP to support 10, 100, 1000 Mbps?

Shannon Capacity **C=Blog2(1+SNR)**

a channel has limited bandwidth in frequency domain

10Base-T (10Mbit/s Base band signal Twisted pair)

Link encoding: Manchester code 1 :up 0: down

advantage: recover clock- get synchronize

disadvantage: very busy--high frequency component very strong(means need more frequency spectrum)

图

Can we extend this to 100Base-T?

10-100①use better cable②changing code scheme(bit rate *10, but in frequency domain it not increase 10 times wider --that would be very nice)-**MLT3**

100Base-T

How to achieve 100 Mbps on the line without significantly increasing the clock frequency?

Multi-Level Transition 3 (MLT3) code

"1"-----1 0 -1 change(3 levels)

"0"--do not do anything

reduce the frequency requirement

do not has DC component---average =0

Bandwidth = **¼ bit rate**

4B/5B encoding for synchronization

(100 * 5/4)*1/4=31.25 MHz--require frequency

if directly use MLT-3, what is the problem? lots of "0"-->not transition--lost clock information (bad for synchronization)

so=>**use 4B/5B** ---narrow band

Power Density Spectrum (PDS)

Synchronization and 4B/5B Encoding

What would be the problem if we convert bit streams to MLT-3 waveform? -If the bits are 00000000000000, the corresponding MLT-3 will be flat! Flat signal→no way to recover the clock to reach synchronization

Solution: 4B/5B

Use 5 bits to represent 4 bits based on the table (left)

This guarantees there would not be long consecutive 0's

▸always use 4B5B--capacity is low(one more bit--but bit rate increase) ▸table look up--hardware- easy to realize ---in receiver, in verse operation is easy

1000Base-T 100-1000 very hard

Use 4 parallel channels (full duplex)

Each channel

-250 Mbps=1000/4 -5-level pulse amplitude modulation **(PAM5)** -{-2, -1, 0, 1 2} -125 Mbaud -Spectrum about the same as MLT-3 -0 used for **FEC** (Frame Error Correction)

Above 1000 base T, even use fiber-- -distance -very short

10G Ethernet **(object different [use SONET to do interconnection])**

10GEthernet---now to interconnect Router

Data Center use 10G to connect sever and switch

-Designed for WAN -Optical fiber based -Copper version newly developed

Compatibility **Frame format** -Keeps the same **Link operation mode** -Does **not** perform auto-negotiation before data exchange

Ethernet over SONET/SDH

Why EoS? **Existing infrastructures** -Ethernet LAN -SONET MAN/WAN **Requirement**-Interconnect LANs within a MAN -WAN access

Protocol Stack 图

SONET- has time slot (TDM) do not care about content- layer 2/3(IP header ,MAC add....)

Things to do...

Mapping

Ethernet is packet-based, SONET is circuit-based.--GFP

A---[X] (buffer)--(TS)--M1--(E1)--M2--(TS)--[Y] (buffer)--B

Preamble(get synchronize)-header-payload

Allocate a timeslot connect X to Y.

Use TDM to carry Packet

Capacity adaptation

Ethernet: 10M, 100M, 1G, 10G

SONET: 51.84M, 155.52M, 622.08M, 2.488G

Bandwidth adjustment

Non-fixed bandwidth provisioning

???: How to support multiple traffic types over the existing transport network infrastructure?

GFP-Generic Framing Procedure -

-a layer 2 telecommunication protocol

Pure SONET -Optimized for voice, not for data -Mapping from data to circuit needed

Problems of existing encapsulation protocols

HDLC--has a flag to tell you the header

ATM--cell delineation

GFP-Generic Framing Procedure

Ethernet frame(already has CRC) + GFP header

Core header--[LEN(CRC) | HEC]

LEN- tell the length of frame;

HEC is calculated base on LEN

Question: What to send during the interval of two packets?

Adding an idle GFP frame will solve the problem. An idle GFP frame contains only the PLI and cHEC, which are both set to 0. Such idle frames help to keep the synchronization and are small enough to avoid wasting bandwidth.

R1--(100Mb/s)--[EoS] ==circuit(51Mb/s) ==[EoS]- -(100Mb/s)-- R2 at [EoS]- -give bottleneck; simple layer 2 device -no retransmission -if just send one packet --it can work -if many packet---it would cause packet lost (buffer overflow)

VCAT-Virtual Concatenation 虚拟级联

Fact: Mismatch between Ethernet & SONET bandwidth

Requirement: Achieve high bandwidth efficiency

Solution: Group a number of circuits for higher bandwidth **Main problem to solve:** Different delay along different circuits

2 different path--bundle them together---delay may be different(SONET need synchronization)

key-measure the delay--delay transmission

LCAS-Link Capacity Adjustment Scheme

Dynamically adjust (increase or decrease)link capacity

Adjustment without traffic loss

Benefit *Bandwidth on demand *Quality of Service *Load balancing *Fault tolerance

Architecture of Line Card 图

Carrier Ethernet: Extending to Metro and Wider Areas

Virtual LAN (VLAN) In this network, all the computers are interconnected using Ethernet switches Without special configuration, they can 'see' each other. We want to make the network looks like three independent LANs: VLAN1, VLAN2, VLAN3

How? host in different VLAN , just use IP to communication

use layer 2 device to connect 2 group

problem(1)Broadcast--ARP/DHCP/Address learning

Security===>separate 2 domains--use layer 3 device(Router/L3 switch) talk to each other through IP

Ethernet not allow Loop--flooding broadcast storm--spanning tree- some switch would be block

In same LAN--just use MAC address

In different VLAN use MAC&IP

IEEE 802.1Q VLAN

Extend the MAC header to include a **VLAN ID**

In a switch, a VLAN ID is associated with certain ports.

[VLAN| DST |Port] MAC address & VLAN ID

Extending to More VLANs

802.1Q supports up to **4096** VLANs only

802.1AD extends this to include two VLAN IDs

Going to Wider Area

Goal: to interconnect multiple Ethernet networks through a backbone so that they look like a LAN

Basic Idea

Use MPLS LSPs or IP tunnels to build a **full mesh**

Ethernet over IP(IP and MPLS can carry every kind of data)

use IP as a vehicle to carry layer 2 frame

What are the Issues?

If we think the core network as a giant switch, what does this switch do?--where to go Why we say MAC address space is flat? IP is hierarchy Why flat address space causes problem? Don't have priority to forward What about? broadcast traffic

Can we avoid broadcast storm? hierarchical -HW

The Big Issue

Scalability：*Each edge router has to know the MAC address of every single computer! *Full mesh would not scale well (very expensive) **Solution**: Use a hierarchical architecture

Provider Backbone Bridges

Access--Aggregation--Core--HW 考虑外层 header

However it reduce the network resilience-- core do not care about source and destination

Lec5 Data Center Networks

Datacenter Elements rack(switch); blade server

Definition of Data Center

A data center is a **facility** that

*houses computer systems and associated components (e.g., switches, routers, storage, power, HVAC)

*provides various services, e.g., storage, processing, exchange, software

Logical View of A Datacenter

Internet=4 **service tier** :**Web, directory, application, storage** (between has Interconnection Network)

query in→(firewall)**Internet**→(firewall)**Web service tier**(not perform search)→ask **Directory service tier**(find a service to search)→respond to **Web service tier**→ **Application service Tier** retreat data (require multiple services to get result)→result back to **Web service tier**

Datacenters: yesterday and today

Traditional datacenters

*Host a large number of small- or medium-size applications

*Each application runs on a dedicated hardware infrastructure

*Different computing systems may have little in common, and may not communicate with each other

***More like a collection of computers**

Current datacenters

*Host a small number of very large applications (e.g., gmail)

*Homogeneous hardware/software platform

*Intensive communications within a datacenter

***More than a collection of computers**

Networking --hierarchy

A Typical Design（图）

Ethernet/IP; Internet ;

Hardware (encryption and decryption);Sever; Storage

UPS(Uninterrupted Power Supply)--**but not last long**

Design Considerations

Scalability

'Scaling out' 横向扩充 instead of 'scaling up'按比例放大: using a large number of low-cost commodity components

Commodity servers and desktop-grade disk drives

Research: using commodity switches to scale out the data center network

Availability

Tier II: 99.7%; Tier III: 99.98%; Tier IV: 99.995%

Fact: with a huge number of commodity components, it is impossible to avoid failures

Solution: design the system to be fault-tolerant

Traffic Engineering

Latency, bandwidth and packet loss are critical

Features of datacenter networks

*Rich interconnections

*Relatively regular topologies

Such features should be leveraged to optimize traffic engineering

*Load balancing*Multipath routing*Reactive flow control

Geo-Distribution

***Placement *Sizing *Network cost *Application design**

Virtualization

Server a physical server hosts multiple virtual machines (VMs) ; A VM can be dynamically created and migrated

Storage : Many physical storage devices are consolidated as a single virtual storage space

Network virtualization: a physical network is sliced to multiple virtual networks, each with specific topology and QoS guarantee

Power Consumption

-When it comes to energy waste, data centers are among the world's biggest offenders. Most physical servers run at only about 10 to 15 percent utilization.

-The unutilized servers in the United States alone emit more carbon dioxide each year than the entire country of Thailand. [Cisco]

Lighting1% transformers/weithchgear1% PDU5% CRAC9% UPS18% Chiller33% IT Equipment 30%

Lec 6 Scheduling

-Multiple users **competing** for shared resource -Scheduling algorithm is used to control and contention by allocating resource among the users according to certain policy -Scheduling is critical for (QoS) provisioning

General Case -A number of packet flows to be statistically multiplexed to a single channel -Each flow has a dedicated queue -A scheduler controls data transfer from the *queues* to the channel

Queue Manager in an Output-Buffered Switch

input link...**switch fabric**--**queue manager**..--output link

packet scheduler 图 come-->store in a piece of memory(queue) **CPU**-(process packet--manage1.know how many queues have here ;each queue has how many packet?2.priority; 3. bandwidth) **Packet search Engine** -Algorithm

Prioritizing User's Traffic

•Why *prioritizing* user's traffic *to meet various quality of service (QoS) requirements *to fully utilize network resources *to relax network designs

•What kinds of priority *delay priority (real-time traffic)

*loss priority (data-type traffic)

•Fairness among virtual channels

*Weighted round robin (bandwidth guarantee)

*Weighted fair queueing (delay bound guarantee)

Fairness-different definition for fairness

-need application background

Max-Min Fairness *N* flows share a link of rate *C*.

Flow *f* wishes to send at rate W(*f*)预想, and is allocated rate R(*f*)实际

▸Pick the flow, *f*, *with the smallest requested rate.*

▸If W(*f*) < *C/N*, then set R(*f*) = W(*f*).

▸If W(*f*) > *C/N*, then set R(*f*) = *C/N*.

▸Set *N = N – 1*. *C = C –* R(*f*).

▸If *N* > 0 goto 1.

Implementation Architecture for Round Robin Scheduling

each flow has dedicated queue(buffer)--period search each queue **VCQ**: Virtual Channel Queue

bandwidths are equal=packet size are equal(ATM)

search empty queue waste time--- **DQ**: Departure Queue

[the number of queue that is not empty; empty-skip it]

1 The packet scheduler at the output of an **ATM** switch has m VCQs and one DQ. 2 Each VCQ storing cells' address has only one VCQ number in the DQ. 3 There can be up to N cells from the switch fabric arriving in each cell time slot. 4 When a cell arriving at an empty VCQ, its VCQ value is inserted to the tail of the DQ. 5 The DQ chooses the head-of-line (HOL) VCQ value and send its HOL cell. 6 As soon as a cell is served, its VCQ is checked if there is any remaining cell. 7 If yes, its VCQ value is inserted to the tail of the DQ. If not, do nothing. 8 Since there can be up to N cells arriving and one cell departing in each time slot, up to N+1 VCQ values can be inserted to the tail of the DQ.

Weighted Round Robin bandwidth guarantee

User *i* is associated with **a weight w(*i*)**

w(i) is an integer, -- why? it's easy for hardware process

Frame-by-frame processing, frame length is the sum of the weights Each user gets a bandwidth proportional to its weight

•Consider a round robin system in which every connection *i*, has an integer weight w_i associated with it. When the server polls connection *i*, it will serve up to w_i cells before moving to the next connection in the round. This scheme is flexible since one can assign the weights corresponding to the amount of service required. The scheme attempts to give connection *i* a rate of

公式：

•A better implementation of WRR operates according to a frame structure. For example, suppose the weights are w_A = 3 and w_B = 7, then AAABBBBBBB → ABBABBABBB

Frame-based implementation without bottleneck

Implementation of WRR

1 The packet scheduler at the output of an ATM switch has m VCQs and one DQ. 2 Each VCQ has a register storing its weight (Wi) and a counter (Ci). 3 The counter keeps track the number of cells sent in a frame. **4 A frame size is the sum of all weight.** Let us assume the frame size is F. 5At the beginning of a frame, each counter (e.g., Ci) is loaded with its weight (Wi). 6 Each VCQ is served in round robin if its Ci is not zero. 7 Upon a cell is served from a VCQ, its Ci value is reduced by one. 8 After F cells have been served, a new frame starts. Go to step 5. 9 If before F cells being served, there is no cells that can be sent, a new frame starts. 10 This scheme has a bottleneck of searching a VCQ with a non-zero counter.

if we use WRR to process packet--segment packet in to fix size----reassemble

Deficit Round Robin (DRR)-it can handle packet of variable size without knowing their mean size.

Problems with WRR •Good for cells, not for packets •Time-consuming non-empty queue searching

DRR -count # of byte! the **minimum weight of quantum** should be at least the size of **longest** packet

•Active list: with non-empty queues •Idle list: empty queues • Only look into the active list

•Take the first queue, add weight •Transmit the backlogged 待办 packet until the weight is less than the size of the waiting packet •If the queue is empty, put into idle list • Otherwise, put to the end of the active list and move to the next active queue

it need to compare the packet size with quantum

Round Robin Scheduling

With bandwidth fairness; **No** delay bound

2 situation -allocated bandwidth same--delay is different =>long delay will hurt the performance

Generalized Processor Sharing (GPS) 图

•Able to achieve strict proportional fair bandwidth allocation •**Not** feasible

▸Bandwidth can be split infinitesimally-as small as possible

▸All the users can be served instantaneously

Weighted Fair Queueing (WFQ) delay bound guarantee

•Three flows with weights: w1, w2, w3

•The total output bandwidth is B

•Flow k gets a bandwidth of **wk*B/(w1+w2+w3)**

WFQ is also called "Packetized Generalized Processor Sharing (PGPS)"

-Flow1 is expected to get bandwidthb1=w1*B/(w1+w2+w3).

-If a packet of size L arrives at time t, the expected transmission time is L/b1

-If the flow has been idle, then the above packet transmission is expected to finish at t+L/b1

-Here t+L/b1 is called virtual finishing time

-Naturally, if we have multiple packets waiting to be transmitted, the scheduler should send them out based on the virtual finishing time stamps.

WFQ: Virtual Finishing Time

For a flow, suppose its allocated rate is R, its jth packet arrives at A(j) and has a size of L(j) , then the virtual finishing time of each packet D(j) is calculated as

D(j+1) = max{D(j), A(j+1)} + L(j+1)/R

GPS

Worst-case Weighted Fair Queueing (W2FQ)

Try to achieve a delay bound

delay < backlog/rate + tolerance.

Use GPS as a reference

At a certain time, only consider the "*should have started*" packets according to GPS

Choose the packet with the minimum "ending time".

lec 7 Buffer Management Overview

Why Buffering *Packet streams are not deterministic, buffers can be used to tolerate bursty arrivals *Contention occurs in statistical multiplexing, buffers can be used for ***short-term contention*** resolution

Buffer: the More, the Better?

(in certain case, it's not good) 图

Queueing theory, *increase queue length* may *reduce loss probability* (e.g. M/M/1, M/D/1)

Buffer Management

Heavy traffic, shortage of buffer

→ what to do upon new arrivals *Drop new arrivals

*Accept new arrivals, discard buffered ones.

Packet dropping sometimes has positive effect*: today's network rely packet dropping for ***congestion control* and ***flow control.*** *if packet dropping-> reduce window size and reduce the transmission rate.*

Ex. *Data Center-RTT is short*

Large buffer - means expensive ; not necessary

*store lots packets - --**long delay**- hurt the performance*

in practice- we need quick reaction!

Packet dropping is necessary for today's Internet. TCP depends on packet dropping to detect network congestion and performs rate control accordingly. Selective dropping distinguishes packet priority and is a fundamental mechanism for class of service. Dropping is also what we need for attack countermeasure where malicious packets need to be discarded.

Buffer Management -- Where to store the arrivals?

(different logic queue)

Scheduling--Which one to send?*(how to departure?)*

Questions

When to drop a packet? *Until buffer overflow *Buffer is near overflow(√) *before overflow(early discard)*

Which packet to drop?

*Only current new arrival*Current new arrival and a certain number thereafter *Those already saved in the buffer

Guidelines

(1)Goodput: High effective message delivery rate(**# of useful message [bits or bytes]**) **ex** IP over ATM, a large IP packet is segmented into multiple ATM cells, dropping any of them results in a corrupted IP packet.

(2)Fairness: If buffer is shared among multiple flows, *small flows* are not starved by *greedy ones*

should make space for small flow **(3)Simplicity:**

Hardware implementation; High speed algorithm

Algorithms

Drop Tail (DT) *Drop arriving cells once the buffer is full

Simple(very straightforward)* *May have poor goodput *No fairness mechanism

Partial Packet Discard (PPD)

*Based on DT *Designed to transport **TCP** segment over ATM *Discard the new cell in case of overflow **Discard all the successive cells belonging to the same TCP segment*** **if belong to same packet --- drop it(generate fewer incomplete packet)** *Goodput improvement over DT *Still some corrupted packets

Early Packet Discard (EPD)

*Set a threshold TH *Once queue size > TH, complete receiving of current segment cells, but discard new segments completely **avoid create partial packet- do not generate incomplete packet** *Goodput improvement over DT and PPD

Performance: ***Goodput: DT< PPD< EPD***

Review: TCP Congestion Control 图

Rule for adjusting W

If an ACK is received: W ← W+1/W

If a packet is lost: W ← W/2

TCP *Feed-back congestion control *Fast feed-back → better performance *Congestion reflected by packet dropping *Early dropping → early feed-back

Drop From Front **detect early**

*Designed to improve TCP performance in case of congestion *Drop the packet from the head of the queue and make room for the new arrival

*Source node is notified one entire buffer sooner

Random Early Detection (RED)

Congestion avoidance: Proactive dropping before congestion really takes place

Synchronization avoidance

>If all the TCP users experience packet dropping, all of them reduce TX rate, link utilization may be low

>*Randomly select* a number of users for dropping

Fairness: Do not introduce bias to bursty flows **(the more you send, the higher probability your packet be dropped)**

Main Ideas: Drop packets *before* buffer full; Drop packets probabilistically

Parameters Thresholds: *minth* and *maxth*

Maximum drop probability: *maxp*

Mechanism *Calculate *avg* (average queue length) at each packet arrival *If *avg* < *minth*, accept the arriving packet * If *minth* < *avg* < *maxth*, drop (or mark) the packet with a probability depending on *avg* *If *avg* > *maxth*, drop the arriving packet

Process A new packet arrives

Check the average queue length

*<min: accept the packet

* [min, max]: discard with a probability *Pb*

*> max: discard

Question: How to determine ***Pb***?

RED Probability Function 公式

Count: # of packets being accepted since the previous packet drop **Pa** is the actually probability used for packet dropping. This ensures an upper bound for non-dropping packet sequence.

For example, we assume Pb=1/10,

*When the 1st packet arrives, count=0, Pa=1/10, suppose the packet is not dropped *when the 2nd packet arrives, count=1, Pa=1/9, suppose it is not dropped *when the 3rd packet arrives, count=2, Pa=1/8, ….*We can see that Pa increases and will become 1 when the 10th packet arrives, which ensures a drop **within** 10 packets.

RED Performance (vs. Drop Tail Queuing Policy)图

the max throughout bound : RED>Drop tail

so RED is widely used than Drop Tail

Lec 8 Flow and Congestion control

Δ: packet transmission time depend on 2 factors:

1. the size of packet 2. the speed of the interface

Flow Control

Data transfer TX → RX, reliable link(in network switches and routers have buffer, we ignore them in this picture)

Speed mismatch between TX and RX

How to realize speed adaptation?

Basic flow control:

(1)Stop-and-Wait Flow Control

Source transmits frame Destination receives frame and replies with acknowledgement Source waits for ACK before sending next frame

What is the major problem?

not efficient ; data rate is small (W=1)

(W=1),in one RTT, we just deliver one packet(the bandwidth utilization is pretty low)

(2)Sliding Window Flow Control

Allow multiple frames to be in transit

if W= infinite---keep sending , keep sending ~~ it means no control!--cause congestion

Receiver has buffer **W** long

(find appropriate window size-bottleneck)

if W=3, in one RTT, we can deliver 3 packet(the bandwidth utilization is triple)

Transmitter can send up to W frames without ACK

Each frame is numbered

ACK includes number of next frame expected

Can we use this over the Internet for end-to-end congestion control? **network is dynamic, W cannot fixed.**

window size should adaptively adjust

Transport Layer Flow Control

Performs end-to-end flow control across the network

Necessary for QoS : **Oct. 1986, first Internet 'congestion collapses': data throughput from LBL to UC Berkeley (sites separated by 400 yards and 2 hops) dropped from 32 Kbps to 40 bps.**

2 types of bottleneck in the Internet:

1.*receiver* has potential bottleneck (constrain at receiver) receiver has buffer size 2.*network* has bottleneck (Router---window size adjustment)--cwnd

TCP credit scheme

Sliding-window * An ACK acknowledges received packets * An ACK also expands transmission window

Credit scheme *Do *acknowledgement* and *window size control* separately *Feedback: ACK + cwnd

TCP Congestion Control

Transmission window is determined by both receiver credit and control window

awnd=min(credit, cwnd)

No congestion: increase window size

Congestion: decrease window size

Congestion implicitly indicated by **packet dropping**.

>>Control of the cwnd (key for TCP)

The congestion window, **cwnd**, determines the transmission rate The key is how to adaptively adjust the cwnd when a congestion occurs

>>Congestion Control

RX buffer is enough

Window size (TX rate) depends on link bandwidth

Abundant bandwidth →packet delivery → ACK → 加 window size

Congestion → packet drop → no ACK → 减 window size

Major TCP Flavors

(1)Old Tahoe 图 **Slow start**(double window size) + **congestion avoidance**(increase linear(by 1))

>>Congestion Avoidance

*Starts when *cwnd ≥ ssthresh*

*On each successful ACK: *cwnd ← cwnd + 1/cwnd*

*Linear growth of cwnd *each RTT: *cwnd ← cwnd + 1*

>>Packet Loss

packet loss----**wait** until timeout

Assumption: loss indicates congestion

Packet loss detected by

*>Retransmission TimeOuts (**RTO** timer)*

>Duplicate ACKs (at least 3) 图

>>problem of Old Tahoe waiting timeout----> waste time!!! --Receive duplicate ACK **4**

(1)most case it implies --packet loss--congestion(if really bad **5 6 7** also drop)

(2)congestion may be not that bad---RED random drop

(2)TCP Tahoe

Slow start + congestion avoidance

Fast Retransmission (do not wait until timeout)

>Wait for a timeout is quite long

>Immediately retransmits after 3 dup ACKs without waiting for timeout

? why not immediately retransmission after 1 dup ACK? Because it may **not** cause by packet loss; it may due to packet **misorder** !

if 3 dup ACKs, the probability that packet loss is high~!

Q1: 1,2,3,4,5,6 are sent out, only 2 is dropped, what happens next using Tahoe?

(hint: cwnd starts from 1, **very slow**)

>>problem of Tahoe： high data rate---> low data rate-- too aggressive (too much)

Q2: Any improvement to Tahoe? ---**react quickly**

(3)TCP Reno ***SS + CA *Fast Retransmission**

***Fast Recovery** >Fast recovery is performed by NOT setting cwnd down to 1. **Rather we cut it to half of the current cwnd**.

>This is based on the assumption that the congestion indicated by duplicate ACKs *is not severe*.

Fast Recovery in TCP Reno

>>At 3 dupACKs Ssthresh = cwnd/**2**

Cwnd = ssthresh+3 Retransmit the lost segment

>>Each additional dupACK

Cwnd ++ ; Transmit a segment if allowed by the cwnd

>>At the next ACK Cwnd = ssthresh ; Exit fast recovery

Question: What if multiple losses occur?

1 lost-- Ssthresh 1 = cwnd/**2**

2 lost-- Ssthresh 2 = cwnd 1/**2**= cwnd/**4**

>>problem of TCP Reno

punish twice---**too aggressive**

(4)TCP NewReno Improvement over TCP Reno

Handles **multiple losses** better

>**Essence**: stay in the **fast recovery** state **until all the outstanding segments are acknowledged**

>Here, **'outstanding segments'** include the segments being previously transmitted but not acknowledged yet

>>At 3 dupACKs, do the same as in Reno

>>At a partial ACK * Retransmit the lost segment immediately * Inflate cwnd based on the new segments being acknowledged

>>At a full ACK *Set cwnd=ssthresh * Exit fast recovery

》》**Problems of Tahoe, Reno, NewReno**

*React to congestion **after** it occurs

*Cannot prevent congestions

we want to do prevention!!!

(5)TCP Vegas

Try to achieve equilibrium during congestion avoidance

Examine TX rate

*Too fast, cause congestion, RTT increases: reduce window size*Too slow, no congestion, RTT does not increase: increase window size *Otherwise: no change

Window size adjustment

Max: *W/RTTmin* Real: *W/RTT*

Difference: *diff = W(1/RTTmin – 1/RTT)*

**diff < a: W++ (this means RTT is close to RTTmin, thus no congestion)* *diff > b: W— (this means RTT is increasing, there could be congestion)* *a < diff < b: no change*

?why it design but people didn't use it?

TCP Vegas vs.TCP Reno

*they start together, **compete***

Vegas aware congestion earlier~

Vegas sacrifice its bandwidth, Reno take more share~!

High Speed + Long Delay

*10 Gbps links*100 ms end to end delay *RTT = 200 ms *Bits in the pipe: 10Gb/s*200ms=2 Gbits *TCP may take thousands of RTTs to *fully utilize* the bandwidth

(RTT is long & bandwidth is high)

--could not fully utilize the bandwidth

Solution: (1)reduce RTT?--distribution system; CDM

(2)start with large window size ~!

but large window size--avg. data rate more high!

-**RED** the probability that your packet be drop is high!

Other Weaknesses of TCP

How about short-life flows?*When multiple flows with different RTTs compete for the same bottleneck link. which has smaller RTT get more bandwidth

Congestion Feedback

Use packet loss

*Long time wait *Congestion? Yes/No how sever?

*Packet loss always means congestion? not accurate

Explicit feedback from routers

*Fast *Accurate: no/moderate/serious congestion(what kind of degree?)

*Adjust TX rate according to the available bandwidth

XCP: eXplicit Control Protocol **Sender:** Include current window size and RTT in packet header **Router** Determine how to adjust window size according to: current window size, RTT, queue length, available bandwidth

Decouple efficiency and fairness computation

Receiver: Feedback the information to sender

Main ideas of XCP :Senders indicate their current window sizes and RTTs in packet headers; Routers perform fair bandwidth allocation among competing flows and compute their window sizes accordingly, which are reflected by updating packet headers; Receivers feedback the collected information back to senders for rate control.

Lec 9 Routing

in a layer 2 domain, just base on the **MAC address**(Ethernet frame header)-ARP request/reply loop-> use spanning tree protocol

AS: in same AS, Router know each other very well, they controlled by same(one) organization.

Packet Forwarding in Routers **Routing table**: [destination, next hop, output port] **Longest prefix matching**: 8.8.8.8 could match 8.8/16 and 8.8.8/24, the router will pick the longest prefix: 8.8.8/24 because it's more specific

Routing Before a router can forward packets, it needs to calculate its **routing table** based on the network information

Routing calculation methods:

>Link-state routing: routers have the entire topology information, e.g., OSPF (open shortest path first)

>Distance vector routing: routers know how to reach a destination, but don't have the entire topology information, e.g., RIP (routing information protocol)

>Path vector routing: similar to distance vector routing, e.g. BGP (border gateway protocol).

***Goal** Find the optimal path from *src* to *dst*

Optimality: delay, congestion, resource usage, etc.

***Approach** *Link/node performance → numeric factors*Evaluate each path *Choose the best path

Routing Table (try netstat –r) 图 which way?

(1) Routing protocol-- find the potential way

(2) prefix matching

ApproachesIn most cases, routing can be mathematically formulated *Heuristic algorithms can be developed based on the formulations

Mathematical Formulation

Network Topology 图

Path 图

Formulation 图

>Minimize

>Subject to

Example 1->3 图

Shortest Path Algorithms

Dijkstra's Algorithm *Building a shortest path tree*Starting from the root node, expand the tree based on link cost*No hop count limit*Link-state routing: each node knows the whole topology each node have ***same view*** of the network and use *same algorithm*

[if have different view of the topology, it may cause loop]

Each router independently calculate its own forwarding table(find the shortest path to all other node)

Router should exchange information

Every single trees **don't** have same shape.

Bellman-Ford Algorithm *Building a shortest path tree *Starting from the root, expand the tree gradually based on the MAX Hop Count *Existing branches can **be modified** in each subsequent step (converge to a stable state) *Distance-vector routing: no need to learn the whole topology

Do not know the real path.

Routing Protocols

Intra- and Inter-Domain Routing

(AS): A group of networks under a common administration and with common routing policies. E.g. an ISP's network

Internet: A number of AS's

Intra-Domain Routing: Inside an AS

IGP: Interior Gateway Protocol

Inter-Domain Routing: Among AS's

EGP: Exterior Gateway Protocol

Example 图

Protocols to Discuss

Routing Information Protocol (RIP)

*Specified in RFC 2453 *Distance-vector routing

*Each router advertises its routing table to its neighbors periodically(slow convergence problem)

*Topology changes can be disseminated within a certain time(Typical interval: 30s)

Scalability *Max metric is 15, max network width is 15.

?Is it possible to increase the scalability by simply setting the max metric to a larger number? E.g. 256.

Theoretically possible, but in practice, it would increase convergence time.

Scalability: Convergence

*Long convergence time may incur upon link failure

Scalability: Solutions **1.Split Horizon:** Do not advertise back to the source node. E.g., A → B, then B does not advertise back to A. But, it cannot handle loops **2.Triggered Updates:** Instead of periodic advertisement, immediate routing table dissemination is performed on topology changes

OSPF Open Shortest Path First (OSPF)*Intra Domain Routing Protocol*RFC 2328*Link-state routing

*The state of each link is propagated throughout the network

*Each router maintains a link state database, and has a picture of the whole topology *All routers have identical database, run identical algorithm and have consistent routing tables

same database + same algorithm= consistent routing table

choose shortest path-- to avoid loop(loop free)

[shortest path ensure loop free;

however, loop free not ensure shortest path]

Comparison with RIP

*Scalability: No limit on the network size

*Overhead: Update triggered only by link state change
*Convergence: Faster
*Multi-Path, Load-Balancing: a router may maintain multiple paths to the same destination router

multiple paths- costs are same- random choose one

OSPF: Link State Advertising *Flooding *Each LSA is flooded to a certain number of interfaces *Each LSA is acknowledged *Upon LSA transmission, a timers is started. Upon timeout (without ACK), retransmission is performed

Two-Layer Hierarchical Routing When network scales up, two-layer hierarchical routing can be employed to **reduce** protocol traffic AS → multiple AREAs
Internal router: inside an area
Area border router: between areas, **condenses** the routing information inside areas, forwards **summaries**
Path cost: intra area + inter area + intra area

BGP Border Gateway Protocol (BGP) *Inter-domain routing *RFC 1171 *One domain does not have the internal information of another domain *Different domains may run different intra-domain routing protocols

The tasks of BGP
>Routing info **exchange** between domains
>Path cost definition
>Decision making of path selection

Routing Information Exchange
Specific **TCP connections** are established for BGP
Four types of messages
Open: confirm the session between two routers
Update: advertise or withdraw routes
Notification: error notification
Keepalive: no routing information, keep connection

Path Cost
A number of path attributes, e.g.
MED: multi-exit discriminator, small is good
Local Preference: large is good
...
Example 图

Path Selection
*When a router is presented with multiple candidate paths, path selection is performed based on the attributes
*The decision making is not defined in the protocol, network administrators can make their own policies

Path Selection: Cisco Example
Prefer large weight
Equal weight: large local preference
Equal: the one from the local BGP speaker
None: shortest AS length
Equal: lowest origin type (IGP < EGP)
Equal: lowest MED
...

Failure Recovery Resume the interrupted services at the earliest possible time
Approaches

***Restoration**
A failure is **detected**
(1)*observe a signal in physical layer-- no signal come in ~
***you monitor the bit error rate --(ex. in link layer -use CRC code)**
(2) Hand Shack--Ping-Acknowledgment
The failure is advertised throughout the entire network
The topology is updated
New paths are calculated
Interrupted services are resumed

***Protection**
When the working paths (primary paths) are computed, backup paths are computed at the same time
Backup paths may **not** carry traffic during normal operation
When a failure is detected, the affected traffic is switched to the backup path(s) **immediately**

***Fast Reroute**
Design for **MPLS** and **IP networks**
Forwarding tables include primary forwarding path and backup forwarding path
When a failure is detected, use the backup forwarding path instead
Very fast, and cost-effective
Usually **no** bandwidth, may cause congestion.

？ keep this hierarchical architecture for routing or not
Hierarchical architectures are easy to build, control and manage. But it's not as flexible as flat architecture. As long as infrastructures are concerns, most likely the future architecture will still be hierarchical.

lec 10 Design of WDM Networks Overview
Generic Network Model
Multi-Fibers between two nodes
Multi-wavelengths in each fiber
multi-fiber --inside each fiber has different wavelength--they run simultaneously=> more bandwidth.
Each wavelength we slice them into
(1)**time slot**----SONET over WDM
(2) packer over wavelength--IP directly over WDM.
Passive Optical Network (PON) .,m
Optical Line Terminal (**OLT**)--【Optical Distribution Network(**ODN**)】--Optical Network Units(**ONU**)
WDM in PON
IP over WDM Architecture
Traffic Routing--
IP Topology Design[**IP topology**]--
Lightpath Routing[**Physical topology** WDM]--

(①Fiber Path Selection or ②Wavelength Assignment)
Service Provisioning 图
Set up **end-to-end** *lightpaths*
Consider wavelength continuity
▸ With *wavelength converters*(different λ)
▸ Without wavelength converters

Wavelength Conversion
O-E-O: Convert ***optical signal*** to ***electrical signal***, then transmit on a desired wavelength
High complexity, high cost, heat dissipation
All-Optical Conversion
Perform conversion in the optical domain, e.g., using semi-conductor optical amplifier (SOA)
High cost, loss of power

Architecture of OXC 图
♦ OXC: optical crossconnect ♦MUX: multiplexer
♦DeMUX: de-multiplexer ♦WC: wavelength converter
no allow same wavelength to same fiber ---**overlap**

Design of WDM Networks
Cost: Use as **few** wavelength converters as possible
Utilization: Set up as many lightpaths as possible
Approach: Consider routing and wavelength assignment

Routing and Wavelength Assignment
Routing: Find the path from the source to the destination
Wavelength assignment
♦Which wavelength on which fiber to use
♦From one hop to the next hop, the switching is constraint by the available wavelength converters

Discuss what challenges are introduced when performing RWA with waveband conversion.
RWA has two components: routing and wavelength assignment. If we have waveband conversion instead of wavelength conversion, the wavelength assignment is no longer flexible. In particular, we have to consider multiple lightpaths as a group to leverage the waveband converters and maximize the resource utilization.

Classification of Service Provisioning
Dynamic Service Provisioning
Connection requests arrives dynamically
Lightpaths are established accordingly

Basics
Service request:
A lightpath from node i to j
Last for a certain period of time
Design Objective:
Setup an available wavelength channel for a request dynamically
Minimize the call blocking rate
Do not disconnect existing connections

Problem Description
Given: a request from node *i* to *j*
Find: a series of links from node *i* to *j*, say, link 1, 2, …, *m*
Each link *k* satisfies:
It has an available wavelength
This wavelength can be connected to the wavelength on link *k*+1

General Procedure
Probe: Source node sends out probing signal to obtain the available resource in the network
Decision Making: Perform routing and wavelength assignment
Ligthpath Setup: Set up the lightpath according to the computation

Routing Subproblem
▸**Fixed Routing**
For a particular [src, dst], find a fixed path that is used for all the lightpaths between them
Low complexity
May result in high blocking rate
▸**Adaptive Routing**
Dynamically find route based on network congestion state
Low blocking probability
On-the-fly computation causes delay
▸**Fixed-Alternative Routing**
Compute a number of fixed routes in advance
Only choose among the fixed routes for a lightpath setup
A trade-off between simplicity and blocking probability

Wavelength Assignment
Random: If multiple wavelengths can be adopted, choose one of them randomly
First-Fit: The wavelengths are numbered, from among all the feasible wavelengths, choose the one with the lowest number
Least-Used: Choose the wavelength that is least used in the whole network
Most-Used:Choose the one that is most used in the whole network

Logical Topology Design
Static Service Provisioning
(**main**-nowadays, prefer for ISP) >IP layer traffic demand is known in advance>Find a way to set up a logical topology on top of the WDM network

Definitions
▸***Physical Topology*** OXCs interconnected by fibers
▸***Logical Topology*** IP routers interconnected by lightpaths ▸***Static Design*** Determine the configuration and do not change it frequently
if change-->reconfigure optical channel-->recalculate IP routing table->instability! not good!
Can tolerate longer computation time

General Problem Description
Given: *An IP layer traffic matrix (measured and/or estimated)*
A physical network
Find: *A optimal/sub-optimal logical topology (what to optimize?)

Objective and Constraints ***Objective:*** Minimize the cost (wavelengths being used) ***Constraints:*** *Routing in the IP layer *Routing and wavelength assignment in the optical layer *Congestion in the IP layer
topology-Ring -heavy congestion

Subproblems
Logical Topology: How to interconnect the routers?
Lightpath Routing: How to route the logical links?
Wavelength Assignment: Which wavelength to use for each logical link?
Traffic Routing: How to carry traffic in the logical topology?

General Approach
1. Based on the traffic demand, determine a logical topology
2. Map the logical topology to the physical topology
3. If step 2 fails (or not good enough), make adjustment of the logical topology and return to step 2

Lec 11 Network Resilience Resilience means the capability of *a network to recovery from failure(s)*
Resilience is evaluated by **several factors**
*Speed: how fast is the recovery?
*Cost: how much resource is needed?
*Type of failure: link/node, single/multiple(80%)
*Complexity

Type of Failures
*Link failure *Node failure
*Single failure*Multiple failure
People pay more attention to **single-link failures** because it has the highest percentage

Recovery: which layer?
Recovery can be done at multiple layers. different layer has its own restoration approach (use them specific application)
Consider IP/MPLS/SONET/WDM
IP: OSPF reconvergence change the topology-recalculate the routing table **RIP**-take long time to converge **MPLS**: LSP protection primary path and back up path **SONET**: automatic protection switching
WDM: wavelength/fiber protection
IP over SONET *a failure occur SONET react very quickly--Router do not react -do nothing
*Router observe failures ->do recovery--SONET device observe nothing

Protection vs. Restoration
Protection
*Allocates spare resources (such as **backup paths**)
*Fast recovery, but low spare capacity efficiency
Restoration
*Computes new network configurations on the fly
*Slow recovery, but high resource efficiency

Protection Schemes for Connection-Oriented Networks WDM, SONET, ATM, MPLS, etc.
connection-oriented: connection before communication
TCP? TCP over IP, IP do not give you a connection, so TCP more like a session not a circuit!
network failure--->retransmission

Typical Schemes:
>1+1 Protection (no way to share)
Two paths established(**working facility** and **protection facility**) Signals are **duplicated** and transmitted through two paths; Receiver picks one of them(compare --pick a normal one)
use twice bandwidth
>1:1 Protection(allow share)
One working path, one backup path
Working path carries data, backup path **idle**
Switch to backup path upon failure
advantage: save recourse--**sharing** compared with 1+1
>1:N Protection
N working paths ***share** the* **same backup path**
Among the N working paths, **only one** could fail at a certain time
Usually these N working paths are physically diverse

Comparison
*1+1 is **faster** than 1:1/1:N because the near end does not change at all, it requires the far end to pick the better signal *1+1 and 1:1 requires considerable resources for backup paths *1:N is more **resource efficient**, but requires careful network design

Path-based Protection
A backup **path** is established between two ends of a path
link1+link2+...=path
Link-based Protection For each **link** a backup path is established for protection.
Comparison >Path-based protection usually gives higher spare capacity efficiency >Link-based protection reacts to failures **faster** **one link failure---neighbor node immediately detect it--react faster**
▸Ring-based APS(not mentioned in class)
▸p-Cycle Protection
*1+1, 1:1 and 1:N belong to linear systems
*p-Cycle provide **ring-like** protection in mesh network
p-Cycle: embed rings in a mesh topology and use such **rings** to protect links or paths.
>p-Cycle: on-cycle protection
the way you want to protect is a part of the ring
>p-Cycle: straddler protection
both end (node) belong to the ring
This ring can shared by multiple link for protection
(node 2 does not in the ring ---should modify the ring or create a new ring)

Resilience Design in IP
♦OSPF Reconvergence
Detect a failure
Advertise the failure throughout the network
Calculate new routes
Update route tables
♦IP Fast Reroute (IPFRR)
Proactively calculate failure recovery schemes and **store** that in the routing table

IP: Shortest Path Routing 图
Consider a *ring*
If A-B fails, theoretically we can still use A-E-D-C
What if A sends an IP packet to E?
The packet will come back to A---because the shortest path from E to B is E-A-B
Question: Is there any way to **'force'** the route A-E-D-C?

IPFRR
Suppose we can **'force'** the route A-E-D-C
If A-B fails, A can simply sends packet through the working route A-E-D-C
MPLS can do that! Because MPLS can use labels to set up an LSP through a desired route
Can we use IP to achieve this? IP-tree-destination base

IPFRR NotVia:
♦Proactive Computation
Give B (the interface being connected to A) a **special** IP address: **BA**
Remove A-B from the topology and find a shortest path from A to BA, say P*
Update the routing table of each node to include P*
key: BA--assign before failure occurs
backup table ---calculate before -update when failure occur
♦Failure Reroute
A detects A-B has failed
For each packet going to B as the next hop, A encapsulate the packet using BA as the destination
This new packet will go through A-E-D-C-B (find the path--only care about reach ability)
At B, the packet is decapsulated, and then forwarded to the real destination.

IP Fast Reroute
Give IP layer fast failure recovery capability that is **independent** of lower layers
Needs modification of routers
Not easy to provide bandwidth guarantee