# Switched Networks

Petr Grygárek

# Layer 2 switching

- eliminates collisions (microsegmentation)
- full duplex improves performance
- hardware-based bridging (high port density)
  - wire-speed switching, low latency

# Requirements to today's switched networks

- Fast convergence
- Deterministic paths
- Redundancy
- Scalability
- Performance of centralized applications, multiprotocol support, multicasting, …

# Redundancy in switched networks

# Problems with redundancy

- Bridging/switching requires topology without loops
  - avoids frame cycling and introducing multiple copies
- Alternative links necessary to implement redundancy
- Need for protocol calculating tree topology
  - Runs between bridges/switches
  - Some ports blocked, other forwarding

# Redundancy implementation options

- L2: Spanning Tree
  - Some links unused all the time (blocked ports)
- L3: Routing protocols
  - Better metric
  - Support for load balancing

# Spanning Tree (802.1d)

- spanning tree is a tree which spans over all switches
- continually monitors network topology and dynamically chooses spanning tree
  - topology is automatically recalculated if some link fails

# Spanning Tree Principle

1. Root bridge election
   - according to preconfigured bridge priorities
   - preemptive
2. Calculation of shortest path tree
   - Shortest paths from elected root bridge to every other bridges
   - preference of individual links may be influenced by configuring link costs
     - by default, cost is inverse proportional to bandwidth
     - if multiple equal-cost path exist, port priority used as tiebreaker
   - Ports on spanning tree are forwarding, other blocking

Both of these phases take place continuously

# Spanning Tree Operation

1. Root bridge election
2. Selection of root port of every bridge
   - forward traffic to root bridge
3. Selection of designated ports for LAN segments
   - necessary when LAN segment attached via multiple bridges
   - path via bridge with lower root path cost is preferred

- Only root and designated ports forward traffic, other ports are blocked
- Ports in blocking state listen for BPDUs
  - transit into Listening state when multiple BPDUs lost

# Mechanics of SPT (1)

- Root bridge generates Bridge Protocol Data Unit (BPDU) every 2 seconds
  - Other switches forward it along spanning tree, accumulating total root path cost inside BPDU
  - Root bridge defines values of various timers
- Every bridge checks for periodic arrivals of BPDUs on it's root port
  - Root port is the port nearest to the root
  - Bridge maintains "best" BPDU heart on every port
    - Link cost added to root path cost when BPDU arrives on port
    - MaxAge timer (20s) used to time-out BPDU

# Mechanics of SPT (2)

- When bridge detects link failure, it reports topology change to the root port (TCN)
  - Every other bridge forwards TCN to the root port
  - When root hears Topology Change Notification, it reports Topology Change in it's BPDUs for some time
    - After bridge hears Topology Change indication, it shortens it's MaxAge timer to MAC address table entries
- To avoid loops during transition to another tree, SPT defines transient states
  - Listening – bridge listens for BPDUs
    - selection of root bridge, root port and designated ports
  - Learning – bridge only learns bridging table
    - still does not forward traffic
    - used to limit flooding
  - Port spends 15 seconds in both Listening and Learning states
  - After link failure, it may take up to 50 secs to build alternative spanning tree
    - After BPDUs lost on some port:
      20 secs to MaxAge timer expiration, 15s in Listening state, 15s in Learning state

# BPDU types/flags

- Topology Change
  - reported upstream (until ACKed)
- Topology Change Ack
  - acknowledges TC to downstream bridge
- Topology Change Notification
  - generated by root when topology change detected

# Spanning Tree Enhancements

Additional features to shorten STP convergence
  - 50 seconds of 802.1d is not enough for today's high availability requirements

- PortFast
  - access ports transits directly to Forwarding state
- UplinkFast
  - fast switch to alternate uplink port when uplink fails (detected by HW)
  - Bypasses Listening and Learning states
  - Mechanism to update bridging table accordingly
- BackboneFast
  - eliminates 20s MaxAge delay when non-directly attached link on root path fails
  - utilizes special BPDU types

# Spanning Tree Security

- BPDU Filter / BPDU Guard
- Root Guard

# Rapid Spanning Tree (802.1w)

- Subsecond convergence (?)
- Backward-compatible with 802.1d (per-port, not recommended)
- Event-based
  - 802.1d Spanning Tree is timer-based
- Proposal/agreement mechanism to make transfer to forwarding state faster
- Implements many 802.1d enhancements
  - PortFast (edge ports)
  - UplinkFast, BackboneFast
- Shared and P2P link types
  - differentiated according to full/half duplex
  - ports on P2P links always forwarding
- Topology change notification flooded directly from bridge which detected it
  - need not to travel to root bridge and back downstream the tree
- Every bridge sends BPDUs independently according to Hello timer
  - does not forward Root bridge's BPDUs as 802.1d
  - BPDUs used as keepalives (3 misses treated as link failure)

# Multilayer switching

# Layer 3 switching

- hardware-based routing
  - speed of switching and the scalability of routing
- layer 3 switch acts on a packet in the same way that a traditional router does
  - not only forwards it but modifies some header fields also (TTL, checksum)

# Routing vs. layer 3 switching

- General-purpose routers
  - microprocessor-based engines
  - software-based packet switching
  - various WAN interfaces
- Layer 3 switch
  - hardware-based packet switching
    - similar procedures as switch applies to frames are applied to packets
  - handle high-performance LAN traffic
  - many ports of homogenous technology (Ethernet)
  - routed or switched port modes
  - VLAN-based, contains virtual router with interfaces to every VLAN
  - If some features are applied, packets must be process-switched, i.e. handled by CPU
    - special kinds of ACLs, per-packet load balancing, …

# Multilayer switching

- Notion of flow (L2, L3, L4)
  - Unidirectional stream defined by IP addresses (and L4 information)
    - May aggregate more TCP/UDP flows between two stations
  - Defined by flow mask
    - Destination IP address
    - Source+Destionation IP address
    - Source+Destionation IP address, L4 protocol, source+destination port
  - Although Destination IP address is sufficient for normal destination-based switching, more specific masks must be used if standard/extended ACLs defined in the network to filter-out unwanted traffic

- First packet routed, following packets of the same flow switched
  - Cache of flow entries maintained
    - Cache entry created when first packet is processed
    - Contains both L3 and L2 information (including input/output VLAN IDs)
    - Used for fast frame/packet header rewrite

# Cisco MLS - multilayer switching implementation

# MLS Components

- Switching Engine (SE)
  - switch with additional intelligence (MLS-enabled)
- Route Processor (RP)
  - router capable to report information to SE

- SE and RP communicate via MLSP protocol
  - Multicasts info about all MAC addresses of RP interfaces (together with LAN IDs)
    - needed by SE to recognize packets sent to the router
  - RP sends flow masks of active flows to SEs
  - Asks SE to invalidate caches when route path changes
    - and when ACLs applied to RP interfaces change

# MLS Operation (1)

- Candidate Packet
  - = Packet sent to MAC address of RP interface (out from local VLAN)
  - Creates entry in SE's switching cache
  - Flow mask associated with switching cache entry defines packet header fields sufficient to identify flow
    - Necessary because of possible ACLs on RP interfaces
    - Flow mask sent to SE by RP via MLSP
- Enable Packet
  - = Packet sent from MAC address of RP interface (to another VLAN)
  - Completes corresponding cache entry

To recognize Candidate/Enable packets, SE must know MAC addresses of all RP interfaces – propagated by MLS

# MLS Operation (2)

- If SE detects packet sent from client to RP interface, it looks for matching cache entry
  - If it is found, packet L3 and L2 header is rewritten, bypassing RP completely
    - SRC MAC=MAC of RP interface, DST MAC, IP TTL--, recalculate IP header checksum
    - Forwards to VLAN according to information in cache entry
      - Based on VLAN membership of outgoing RP interface
- If routing tables or ACLs change on the RP, cache in SE has to be completely invalidated
  - RP reports that to SEs via MLSP
  - in some situations may introduce significant inefficiency

# Cisco Express Forwarding

# Cisco Express Forwarding

- Used to limit searches in the routing table, recursive routing table lookup and ARP-cache entry access needed for frame header rewrite
- Except routing table, router also maintains
  - Forwarding Information Base (FIB)
  - Adjacency Table

# CEF Forwarding Information Base and Adjacency Table

- Forwarding Information Base (FIB)
  - routing table transformed to 4-level 256-way tree for faster searching
    - leaf used to forward particular packet selected on longest-match
  - created in advance (by software process)
    - takes into account all destinations in routing table, not only active flows
    - recursive lookups resolved
- Adjacency table
  - Maintain L2/L3 neighbor information
  - Used to make ARP table search process faster
  - Allows fast packet rewrite

- Nodes of FIB (256-way tree) point into Adjacency table
  - "mtrie" data structure – leafs contain pointers to Adjacency table
  - Not needed to search in additional ARP table
  - Support for load balancing (multiple pointers to Adjacency table used in round-robin way)
  - If L2 neighbor information change, FIB structure need not to be completely invalidated

# CEF Advantages

- Suitable even for many short-term flows (Internet backbone)
- All packets (including first one) routed by hardware
  - HW accesses FIB and Adjacency table
- If topology chances, only part of FIB can be selectively modified
  - doesn't clear complete cache as MLS does

# VLANs

# Earlier and today's LAN traffic characteristics

- "Clasical" 80/20 rule:
  - 80% of traffic remains local (VLAN boundary)
  - Users don't cross backbone to access most of required resources (workgroup servers)
  - Users grouped logically
- 80/20 rule no longer valid for global services
  - 20/80 rule
  - Responds to trend of resource centralization
    - (server farms, Web services, ...)
  - Requires high-performance L3 switching

# VLAN models

- 1 VLAN = 1 IP subnet
- VLAN models
  - campus-wide LANs (older)
    - “end-to-end VLANs”
    - VLANs group devices according to functionality (common resources)
    - regardless to physical location
    - Common security policy for all VLAN members
    - User§s mobility limited to VLAN
    - 80/20 rule
  - local VLANs (modern approach)
    - Uses L3 switches
    - Uses cluster as design unit
    - cluster contains multiple access switches +L3 switch
      - VLANs terminated on L3 switch
    - fast routing (L3 switching) between interconnected L3 switches
    - 20/80 rule

# VLAN membership

- Static (port-based)
  - (non-trunk) ports with no explicit VLAN assignment fall into default VLAN
- Dynamic (based on L2/L3 information, most often source MAC address)
  - VMPS server (Cisco proprietary)
    - Maintains database of MAC-to-VLAN assignments
      - Devices not listed may fall to „default“ VLAN
    - VLAN assignment defined by first frame received
    - Cisco: all devices on the port must be in the same VLAN
      - If following frames should belong to another VLANs, they are not permitted
    - Port-to-VLAN assignment deleted if port is disconnected
      - (linkbeat pulses fail to arrive)
    - Port limitations for users may be defined
      - Specific device may be denied to assign VLAN if connected to wrong port
    - VMPS server: high-end switch (or Linux :-) )
    - VMPS client: has IP address of VMPS server and ports allowing dynamic membership configured
    - UDP-based VLAN Query Protocol (VQP) between VMPS client(s) and VMPS server

# VLAN - Trunking

- 802.1q header (802.1p/q)
  - "internal tagging" – imposition of tag modifies original frame
  - EtherType=0x8100
  - 2B header: 12b VLAN ID, 3b priority (QoS)
  - Maximum frame length extended from 1518 to 1522 B
    - including CRC
- 802.1q trunks also allows untagged frames
  - "Native VLAN"
  - Must be mapped to the same VLAN at both trunk link sides ends to avoid unwanted pass between VLANs
- Trunk configuration may limit VLANs allowed to transit
  - Limits unnecessary broadcasts/flooding to switches where no members of particular VLAN reside
  - May be limited dynamically by (proprietary) pruning protocol (Cisco: VTP) reporting active VLAN members on individual switches
    - VLAN carrying service protocols (STP, VTP, …) never pruned

# VLAN Tunneling

- dot1QinQ
- Encapsulates 802.1q frame in provider network with another 802.1q frame
- Enables overlapping VLANs of multiple separated customers to be transported via VLAN-based provider's core network

# Spanning tree and VLANs

- 802.1q: Common Spanning Tree (CST)
  - 802.1d in VLAN 1, common tree for all VLANs
- Per-VLAN Spanning Tree (Cisco)
  - Separate tree for every VLAN
  - Infrastructure may be used efficiently
  - Requires lot of CPU processing if many VLAN are active
- Multiple Spanning Tree (802.1s)
  - Based on Rapid Spanning Tree (802.1w)
  - Multiple SPT instances, each instance computes SPT for group of VLANs

# Routing between VLANs

# Routing between VLANs

Connection of one VLAN to one router port inefficient for tens of VLANs, there exists better solutions:

- Route-switch module of legacy (L2) switch
- VLAN virtual interfaces on L3 switch
- Router-on-the stick
  - Trunk link between switch cluster and single router
  - Limited scalability
  - For 80/20 traffic pattern, router should be STP root for all VLANs
    - Ensures shortest paths from every switch to the router

# Other issues of switched networks

# Channel Bundling

- Group of ports configured to act as single link and share load
  - Group forms single link from STP perspective
  - Only lines functional at each instant are uses
  - Can group both trunk and access links (all of the same type)
    - Ports of a group have to have identical parameters
    - Speed, duplex, trunking mode, …
  - Various load balancing methods
    - per-source, per-destination, combination of both
    - each side may use different method
    - important to set properly to reach equal utilization of bundle lines
  - One switch may define multiple port groups
    - Groups differentiated by group ID

- Link Aggregation Control Protocol (LACP) – 802.1ad
  - Negotiates port bundles

# Monitoring of Switched Network

- Switched Port Analyser (SPAN)
  - frames received/transmitted by specified port(s) are copied to port designated as SPAN port
- VSPAN = SPAN using VLANs as monitored source
  - copies all frames of particular VLAN to SPAN port
- RSPAN = remote monitoring using SPAN port
  - SPAN port of switch different from the one with monitored ports may be used
    - (switches connected together with trunk link)
  - reserves one VLAN on the trunk link to carry monitored traffic

Due to SPAN port speed limitation, the mechanism is not sufficient for wire-speed traffic monitoring.

# Switched network security

- ACLs supported by hardware
  - Port ACLs
    - In/out
    - Filters according to MAC, but sometimes also IP/Layer4 address
  - VLAN ACLs
    - directionless, define traffic allowed to pass through particular VLAN
  - L3 switch: virtual router interface's ACLs
- Port security
  - Limits (source) MAC addresses allowed on port
  - Limits number of MAC addresses dynamically learned on particular port
- Protected ports
  - disallows to forward traffic to other protected ports
    - restrict peer-to-peer communication

# Building blocks of switched networks

- Switch block
- Collapsed Core
- Dual Core
- Pros and cons of L2 and L3 backbones

# LABs

- Observing STP behavior
  - Configuring EtherChannel
- Inter-VLAN routing with external router
- Inter-VLAN routing with L3 switch
  - + another routed ports
- [Dynamic VLANs (VMPS)]