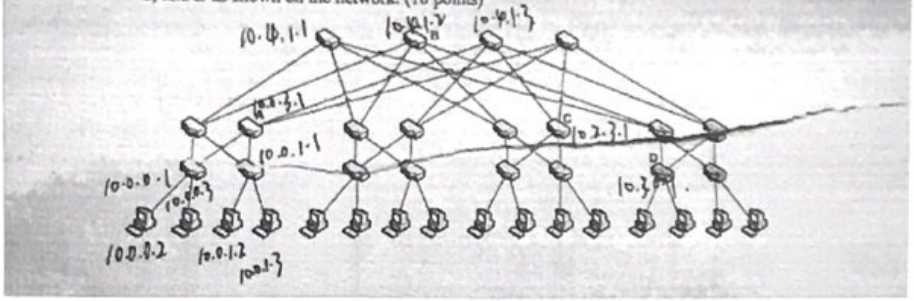
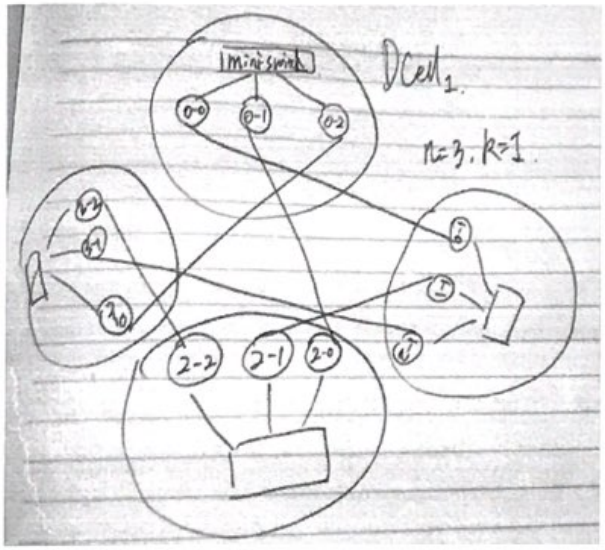
1. Which one is NOT the desired property of the datacenter network? (5 points)
   (1) Scalable interconnection bandwidth;
   (2) Revolutionary clean-slate architecture;
   (3) Economies of scale;
   (4) Backward compatibility

2. According to the addressing scheme of FatTree, give the IP addresses for the switches of A, B, C, and D as shown on the network. (10 points)



A: 10.0.3.1
B: 10.4.1.2
C: 10.2.3.1
D: 10.3.0.1

3. Draw a DCell network structure with n=3, k=1, and name each node.(10 points)



4. How many layers are there in a software defined network? Where is the OpenFlow located? List at least three OpenFlow flow entry actions, and for each action, explain how the OpenFlow switch will handle a packet that matches the entry? (15 points)

1. Three layers
   - application layer
   - control layer

- infrastructure layer

2. Between the control layer and the infrastructure layer
3. three basic actions:
   - Forward this flow's packets to a given port (or ports).
   - Encapsulate and forward this flow's packets to a controller.
   - Drop this flow's packets.

如果匹配OpenFlow flow entry，并且action是转发，则按照entry转发该flow，如果action是drop，则drop该flow，如果不匹配entry，则压缩并转发至控制器，一般情况是转发该flow的第一个packet，也有转发所有packet的情形。

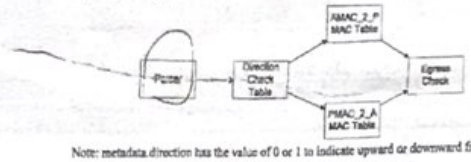
这个第四问的最后一个解释

5. Fill the blanks in the script written in the Ethane policy language, so that:
   Rule 1: Members in "students" cannot communicate with each other; (5 points)
   Rule 2: Members in "students" can communicate with members in "tutors" using protocol of SSH; (5 points)
   Rule 3: Members in "tutors" can communicate with each other; (5 points)
   Rule 4: Members in "students" can use HTTP through the "http-proxy" device. (5 points)

```
# Rule 1
[(usrc=in("students")^(udst=in("students"))] : deny;
# Rule 2
[(usrc=in("students")^(protocol="ssh")^(udst=in("tutors"))] : allow;
[(usrc=in("tutors")^(protocol="ssh")^(udst=in("students"))] : allow;
# Rule 3
[(usrc=in("tutors")^(udst=in("tutors"))] : allow;
# Rule 4
[(usrc=in("students"))^(protocol="http)] : waypoints("http-proxy");
```

6. In PortLand, edge switches are responsible for translating AMAC to PMAC for the upward Ethernet frames and translating PMAC to AMAC for the downward Ethernet frames. We seek to implement PortLand with P4, following is the flow control chart, please fill the blanks in the amac_to_pmac and pmac_to_amac action specifications, and write out the skeleton of the other parts of the P4 program, including the packet parser, the table specification, and the control program. (25 pt.)

Flow control chart:



Note: metadata.direction has the value of 0 or 1 to indicate upward or downward frames respectively.

Action specifications

```
action set_direction(direction){
    //Set the metadata direction
    set_field(metadata.direction, direction);
}
```

```
action amac_to_pmac(ethernet, src_pmac, dst_pmac, egr_spec){
    //Translate the address
    copy_field(_____);
    copy_field(_____);
    //Set the destination egress port
    set_field(metadata.egress, egr_spec);
}
```

```
action pmac_to_amac(ethernet, src_amac, dst_amac, egr_spec){
    //Translate the address
    copy_field(_____);
    copy_field(_____);
    //Set the destination egress port
    set_field(metadata.egress, egr_spec);
}
```

```
action_amac_to_pmac(…){
    …
    copy_field(ethernet.src_adrr,src_pmac)
    copy_field(ethernet.dst_adrr,dst_pmac)
    …
}
action_pmac_to_amac(…){
    …
    copy_field(ethernet.src_adrr, src_amac)
    copy_field(ethernet.dst_adrr ,dst_amac)
    …
}
```

Packet Parser（不一定对）

```
parser start {
    ethernet;
}

parser ethernet {
    switch(ethertype) {
        case 0x8100: vlan;
        case 0x9100: vlan;
        case  0x800: ipv4;
        // Other cases
    }
}
```

```
parser vlan {
    switch(ethertype) {
        case 0xaaaa: mTag;
        case  0x800: ipv4;
        // Other cases
    }
}

parser mTag {
    switch(ethertype) {
        case 0x800: ipv4;
        // Other cases
    }
}
```

Table Specification（不一定对）

```
table mTag_table {
    reads {
        ethernet.dst_addr : exact;
        vlan.vid : exact;
    }
    actions {
        // At runtime, entries are programmed with params
        // for the mTag action.  See below.
        add_mTag;
    }
    max_size : 20000;
}

table source_check {
    // Verify mtag only on ports to the core
    reads {
        mtag : valid; // Was mtag parsed?
        metadata.ingress_port : exact;
    }
    actions { // Each table entry specifies *one* action

        // If inappropriate mTag, send to CPU
        fault_to_cpu;

        // If mtag found, strip and record in metadata
        strip_mtag;

        // Otherwise, allow the packet to continue
        pass;
    }
    max_size : 64; // One rule per port
}
```

```
table local_switching {
    // Reads destination and checks if local
    // If miss occurs, goto mtag table.
}

table egress_check {
    // Verify egress is resolved
    // Do not retag packets received with tag
    // Reads egress and whether packet was mTagged
}
```

Control Program

```
control main() {
    // Verify mTag state and port are consistent
    table(source_check);

    // If no error from source_check, continue
    if (!defined(metadata.ingress_error)) {
        // Attempt to switch to end hosts
        table(local_switching);

        if (!defined(metadata.egress_spec)) {
            // Not a known local host; try mtagging
            table(mTag_table);
        }

        // Check for unknown egress state or
        // bad retagging with mTag.
        table(egress_check);
    }
}
```

补充（不用写）：

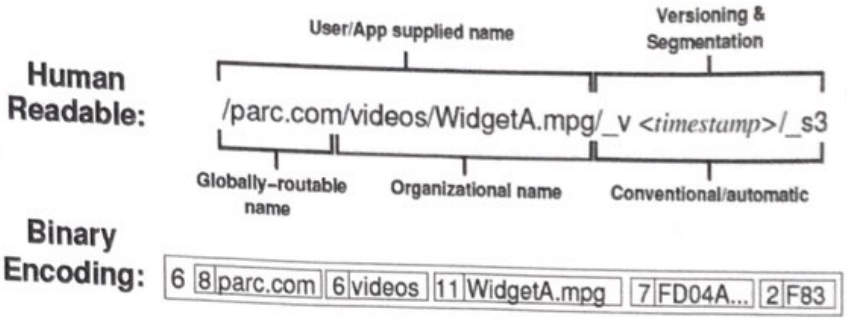
```
header ethernet {
    fields {
        dst_addr : 48; // width in bits
        src_addr : 48;
        ethertype : 16;
    }
}

header vlan {
    fields {
        pcp : 3;
        cfi : 1;
        vid : 12;
        ethertype : 16;
    }
}
```

```
header mTag {
    fields {
        up1 : 8;
        up2 : 8;
        down1 : 8;
        down2 : 8;
        ethertype : 16;
    }
}
```

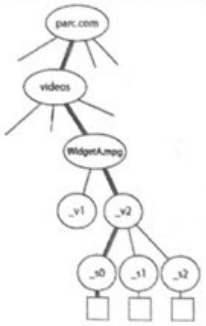
7. How CCN names a content Data? On CCN network, when you receive a Data with a name of "cn/edu/ustc/cs/tian/exampaper.doc/v2/page1/line10", how can you validate the data? (15 points)

Transport: Naming

- CCN is based on hierarchical, aggregatable names at least partly meaningful to humans
- The name notation used is like URI



Transport: Naming



- An Interest can specify the content exactly
- Content names can contain automatically generated endings used like sequence numbers
- The last part of the name is incremented for the next chunk (e.g. a video frame)
- The names form a **tree** which is traversed in preorder
- In this way, the receiver can ask for the next *Data* packet in his *Interest* packet