Inter-domain cooperative DDoS protection mechanism

draft-nishizuka-dots-inter-domain-mechanism-01

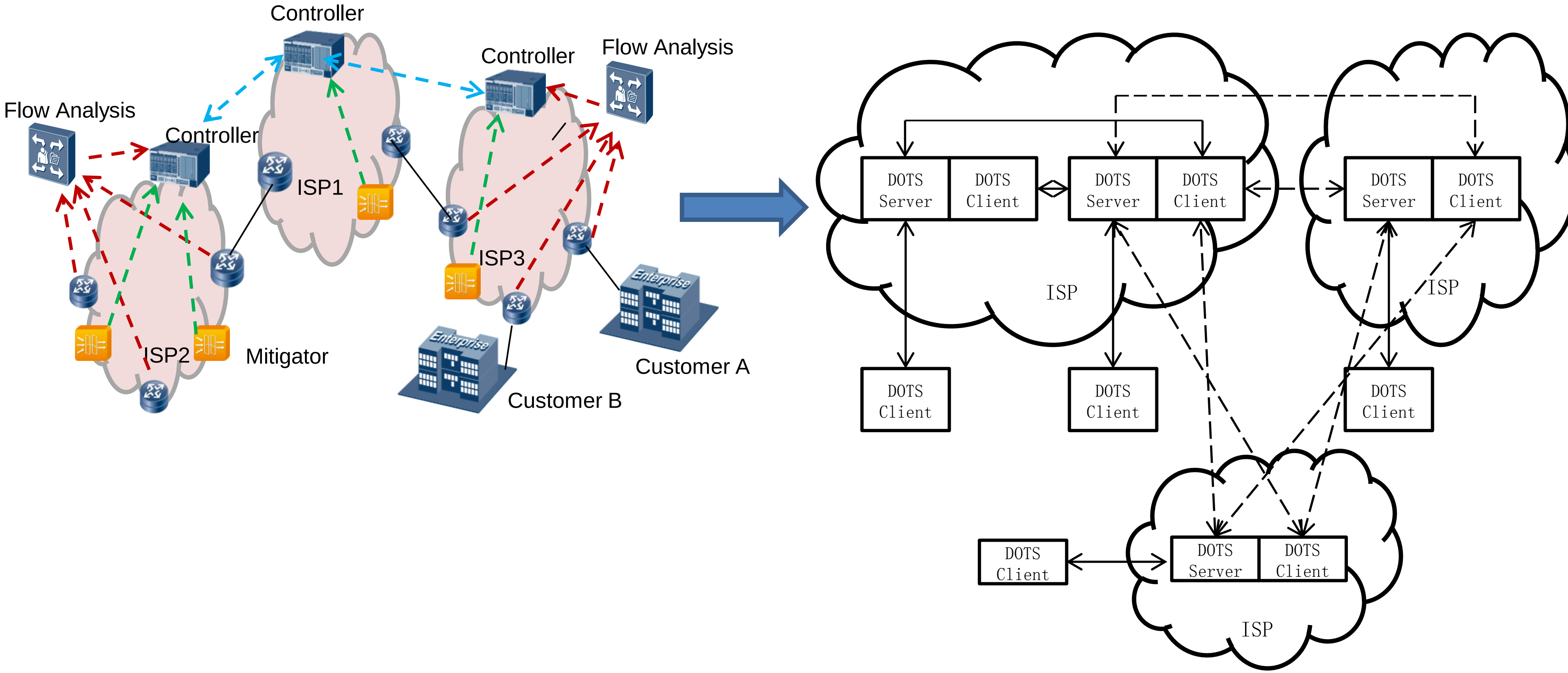
Kaname Nishizuka	NTT
Liang Xia	Huawei
Jinwei Xia	Huawei
DaCheng Zhang	Alibaba
Luyuan Fang	Microsoft
Christopher Gray	Comcast
Rich Compton	Charter

July 2016 Berlin

From -00 to -01

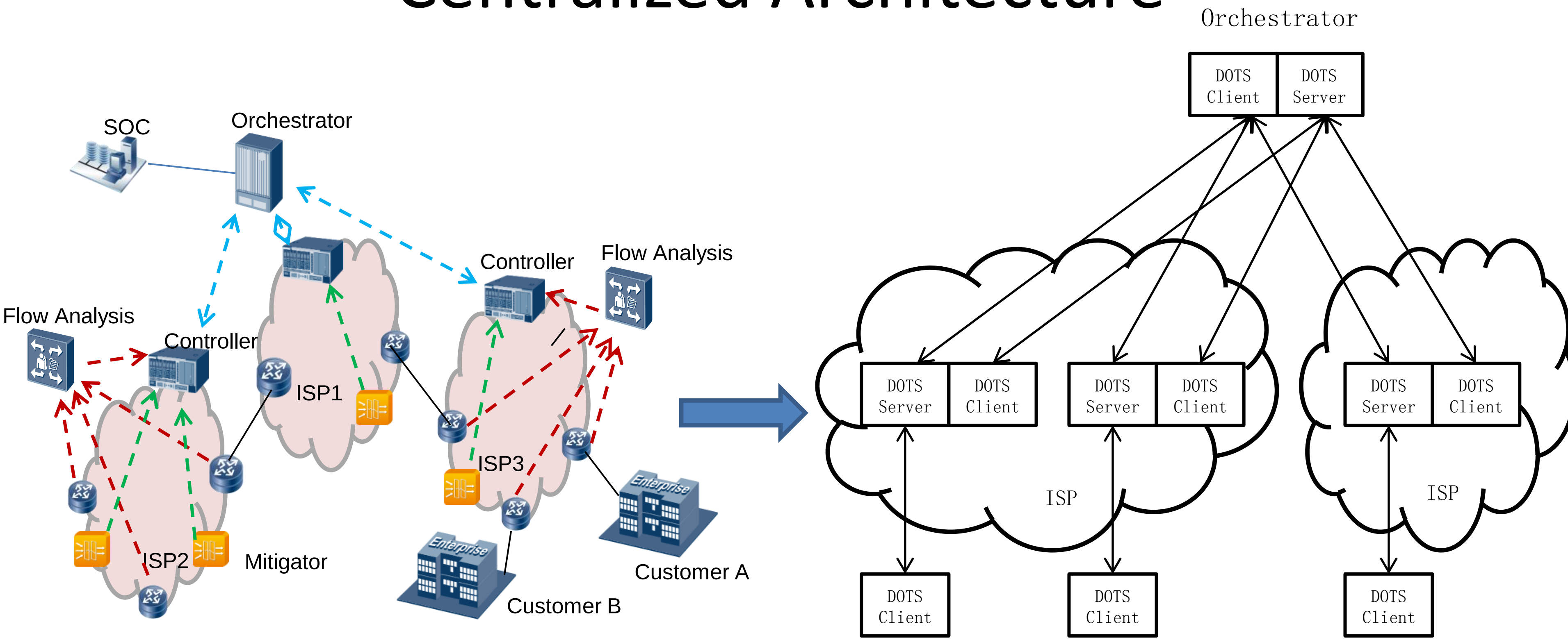
1. Add contents to explain the protocol and signaling messages specification applies both intra-domain and inter-domain situations;
2. Restructuring the contents of Cooperative DDoS Protection Requirements
   - Provisioning Requirements: registering messages for Automatic Provisioning;
   - Coordination Requirements: mitigation request, status exchange, near source mitigation for inter-domain attacks;
   - Returning Path Requirements: routing loops prevention.
3. Redesign DOTS signaling messages and their detailed attributes, as well as the protocol operations;
4. A lot of editorial text changes;
5. New co-authors from Comcast and Charter.

Distributed Architecture



- *Peer-to-peer coordination;*
- *customer<->DOTS client, ISP controller<->DOTS server + DOTS client;*
- *The inter-domain coordination can be a repeated process;*
- *A straightforward and simple solution for the DDoS protection cooperation among small number of ISPs:*
  - ✓ *The incomplete information may not lead to the most optimized operation;*
  - ✓ *Configurations become more complex and error prone as the number of ISPs increases;*
  - ✓ *By repeated coordination among multiple ISPs, It may take a long time to enforce the mitigation.*

Centralized Architecture



- *the centralized orchestrator is the core component to the inter-domain system;*
- *customer<->DOTS client, ISP controller<->DOTS server + DOTS client, orchestrator<->DOTS server + DOTS client;*
- *The inter-domain coordination is bridged by the orchestrator;*
- *Comparing to distributed architecture:*
  - ✓ *The orchestrator has the HA problem;*
  - ✓ *Centralized way facilitates the automatic provisioning of DDoS protection resource and comprehensive information for overall optimized mitigation;*
  - ✓ *Direct communication with orchestrator guarantees quick and fixed DDoS response time.*

Inter-domain DDoS Protocol

- Secure channel (signaling, data):
  - Requirements: confidentiality, integrity and replay attack protection;
  - Mutual authentication: bidirectional certificate authentication ([ITU-T X.509]), unidirectional certificate authentication on the DOTS server, bidirectional digital signature authentication;
  - Solution in this draft: https + JSON;

- Specification for protocol and messages (no difference for all architectures):
  - Provisioning stage
  - Signaling stage
  - heartbeat message:

Provisioning Stage Protocol

- Registration process: facilitate the auto-discovery and capacity negotiation between the DOTS client and server;
  - Messages over DOTS data channel (TLS transport is recommended): registration, registration response, registration cancelling, registration cancelling response;
  - Operations: The DOTS client registers (or cancels registration) to the DOTS

registration body:

```
 {
 "customer_name": string;
 "ip_version": string;
 "protected_zone": {
    "index": number;
    "need_alias": string;
    "ipv4_CIDR": string;
    "ipv6_address": string;
    "BGP_route": string;
    "SIP_URI": string;
    "E164_number": string;
    "DNS_name": string;
 }
 "protected_port": string;
 "protected_protocol": string;
 "countermeasures": string;
 "tunnel_information": string;
 "next_hop": string;
 "security_profile": {
    "TLS": string;
    "DTLS": string;
    "CoAP": string;
 }
 "white_list": {
    "name": string;
    "sequence_number": string;
    "source_ip": string;
    "destination_ip": string;
    "source_port": string;
    "destination_port": string;
    "protocol": string;
    "length": string;
    "TTL": string;
    "DSCP": number;
    "ip_flags": number;
    "tcp_flags": number;
```

```
"black_list": {
      "name": string;
      "sequence_number": string;
      "source_ip": string;
      "destination_ip": string;
      "source_port": string;
      "destination_port": string;
      "protocol": string;
      "length": string;
      "TTL": string;
      "DSCP": number;
      "ip_flags": number;
      "tcp_flags": number;
   }
  }
```

registration response body:

```
 {
 "customer_name": string;
 "customer_id": string;
 "alias_of_mitigation_address": {
    "index": number;
    "alias": string;
 }
 "security_profile": string;
 "access_token": string;
 "thresholds_bps": number;
 "thresholds_pps": number;
 "duration": number;
 "capable_attack_type": string;
 "registration_time": string;
 "mitigation_status": string;
 }
```

registration cancelling body:

```
{
"customer_id": string;
"reasons": string;
}
```

registration cancelling response body:

```
{
"customer_id": string;
"result": string;
}
```

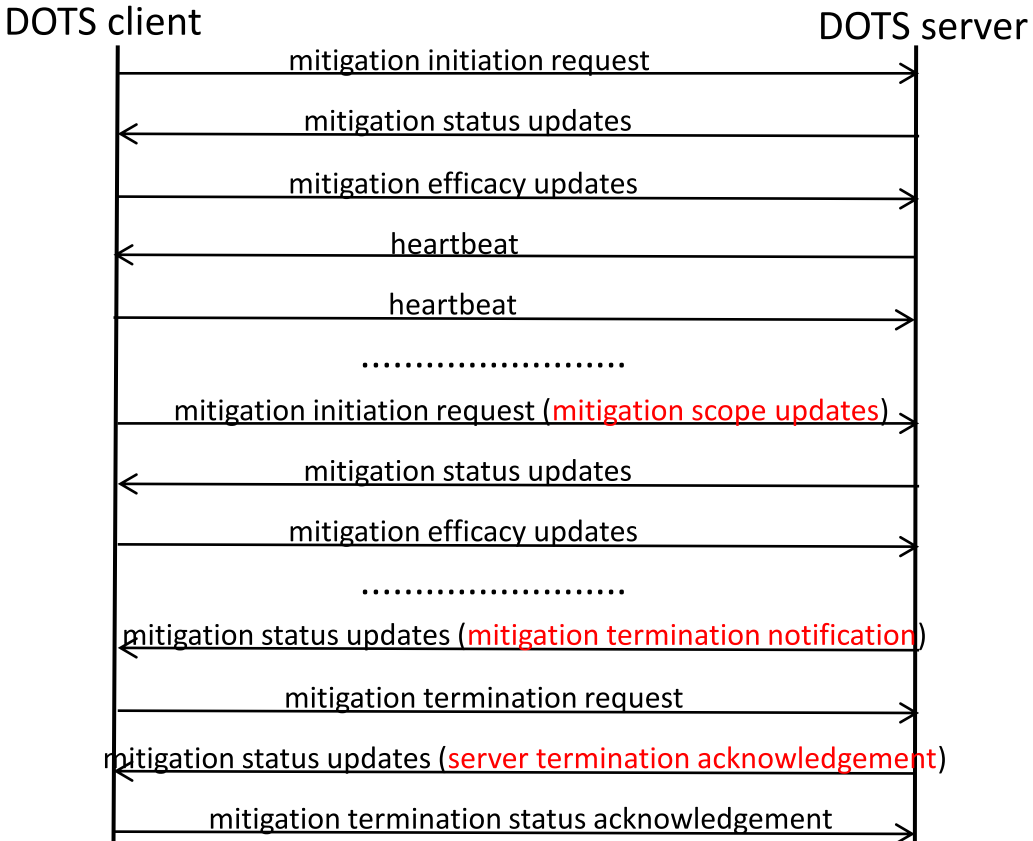
The DOTS server indicates the result of processing the POST request using HTTP response codes:

- Success: Response code 200 (OK) ;
- Fail: Response code 400 (Bad Request) or Response code 500 (Invalid query) with:

  "error_reason": number;
  - 0: Bad Request;
  - 1: Invalid Query;
  - 2: Server Error;
  - 3: Protected Zone Confliction;
  - 4: Countermeasure Not Supported;
  - 5: Security Profile Not Supported;
  - 6: Confliction Exists for White-list or Black-list;
  - 255: Others;

Signaling Stage Protocol

- During DDoS attack: mitigation service request and status exchange over DOTS signaling channel under link saturation;
  - Messages **(**asynchronous**):**
    - DOTS client to server: mitigation initiation request, mitigation efficacy updates, mitigation termination request, mitigation termination status acknowledgement, heartbeat;
    - DOTS server to client: mitigation status updates, heartbeat.
  - Operations:



DOTS client

DOTS server

mitigation initiation request

mitigation status updates

mitigation efficacy updates

heartbeat

heartbeat

....

mitigation initiation request (mitigation scope updates)

mitigation status updates

mitigation efficacy updates

....

mitigation status updates (mitigation termination notification)

mitigation termination request

mitigation status updates (server termination acknowledgement)

mitigation termination status acknowledgement

Signaling Stage Protocol

DOTS client to server

mitigation request body:

```
{
 "version": string;
 "type": string;
 "alert_id": string;
 "sender_id": string;
 "sender_asn": string;
 "mitigation_action": number;
 "lifetime": number;
 "max_bandwidth": number;
 "packet_header": {
    "dst_ip": string;
    "alias": string;
    "dst_ports": string;
    "src_ips": string;
    "src_ports": string;
    "protocols": string;
    "tcp_flags": string;
    "fragment": string;
    "pkt_len": string;
    "icmp_type": string;
    "icmp_code": string;
    "DSCP": string;
    "TTL": string;
 }
 "current_throughputs": {
    "bps": string;
    "pps": string;
 }
 "peak_throughputs": {
    "bps": string;
    "pps": string;
 }
 "average_throughputs": {
    "bps": string;
    "pps": string;
 }
 "info": {
    "attack_types": string;
    "started": number;
    "ongoing": number;
    "severity": number;
    "direction": number;
    "health": number;
 }
"vendor": {
        "name": string;
        "version": string;
        "payload": {
            "offset": number;
            "content": string;
            "hash": string;
        }
    }
}
```

mitigation efficacy updates body:

```
{
 "version": string;
 "alert_id": string;
 "sender_id": string;
 "sender_asn": string;
 "attack_status": string;
 "health": number;
}
```

mitigation termination request body:

```
{
 "version": string;
 "alert_id": string;
 "sender_id": string;
 "sender_asn": string;
}
```

mitigation termination status acknowledgement body:

```
{
 "version": string;
 "alert_id": string;
 "sender_id": string;
 "sender_asn": string;
}
```

heartbeat body ...

DOTS server to client

mitigation status updates body:

```
{
 "version": string;
 "alert_id": string;
 "sender_id": string;
 "sender_asn": string;
 "status": number;
 "error_reason": number;
 "lifetime": number;
 "source_ports": string;
 "destination_ports": string;
 "source_ips": string;
 "destination_ip": string;
 "TCP_flags": string;
 "start_time": number;
 "end_time": number;
 "forwarded_total_packets": number;
 "forwarded_total_bits": number;
 "forwarded_peak_pps": number;
 "forwarded_peak_bps": number;
 "forwarded_average_pps": number;
 "forwarded_average_bps": number;
 "malicious_total_packets": number;
 "malicious_total_bits": number;
 "malicious_peak_pps": number;
 "malicious_peak_bps": number;
 "malicious_average_pps": number;
 "malicious_average_bps": number;
 "record_time": string;
}
```

heartbeat body

```
{
 "version": string;
 "sender_id": string;
 "sender_asn": string;
}
```

Next Steps

- Comments are welcome

- Keep on improving, including:
  - More details about DOTS messages specification, and the protocol operation process;
  - More descriptions about secure channel (authentication, authorization, privacy), transport mechanism.

Thanks!

Liang Xia (Frank)