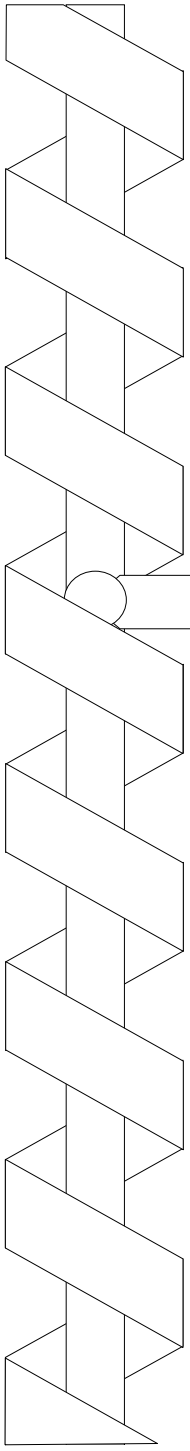


Internet Quality of Service

Weibin Zhao

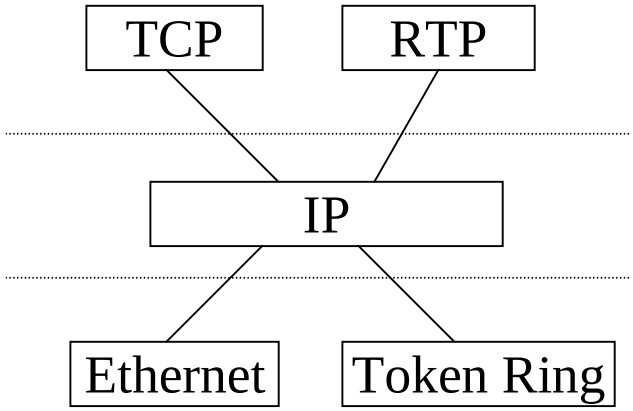
zwb@cs.columbia.edu

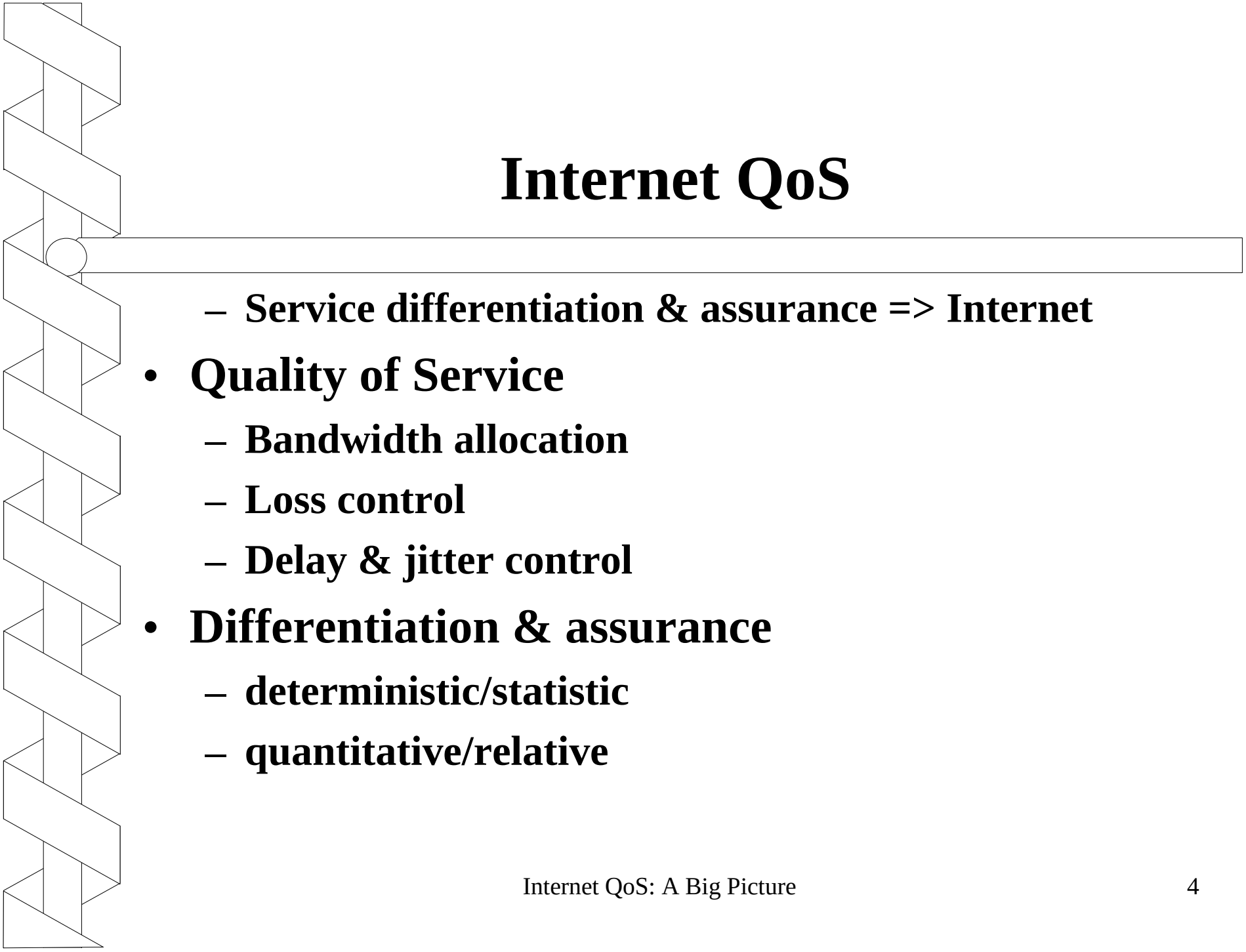
Outline

1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**

Background: IP

- **Characteristic**
  - **Connectionless**
  - **Building block: datagram**
- **Goals**
  - **Multiplexing**
  - **Survivability**
  - **Multi-ToS**
  - **Variety of networks**





Internet QoS

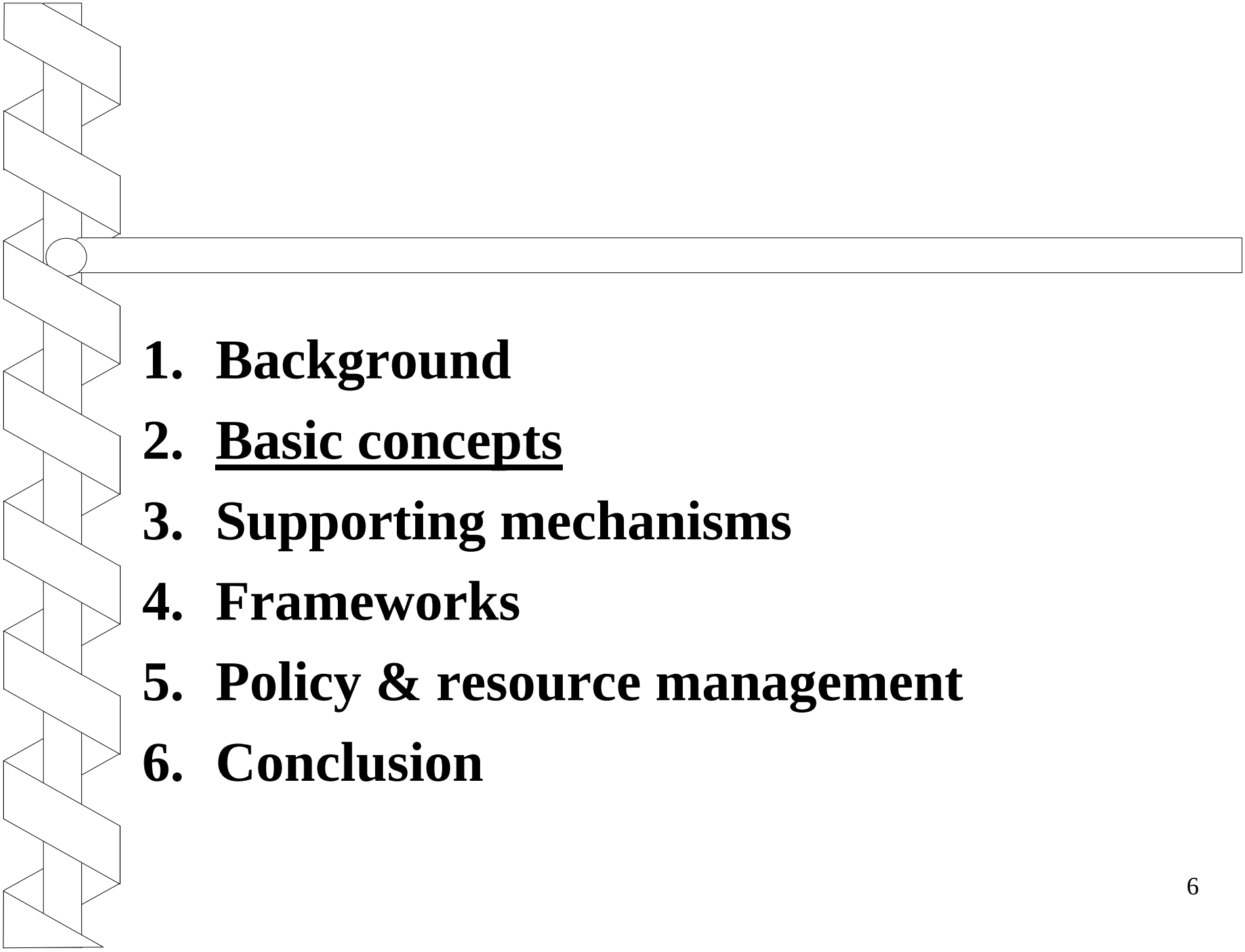
- Service differentiation & assurance => Internet

- **Quality of Service**
  - Bandwidth allocation
  - Loss control
  - Delay & jitter control
- **Differentiation & assurance**
  - deterministic/statistic
  - quantitative/relative

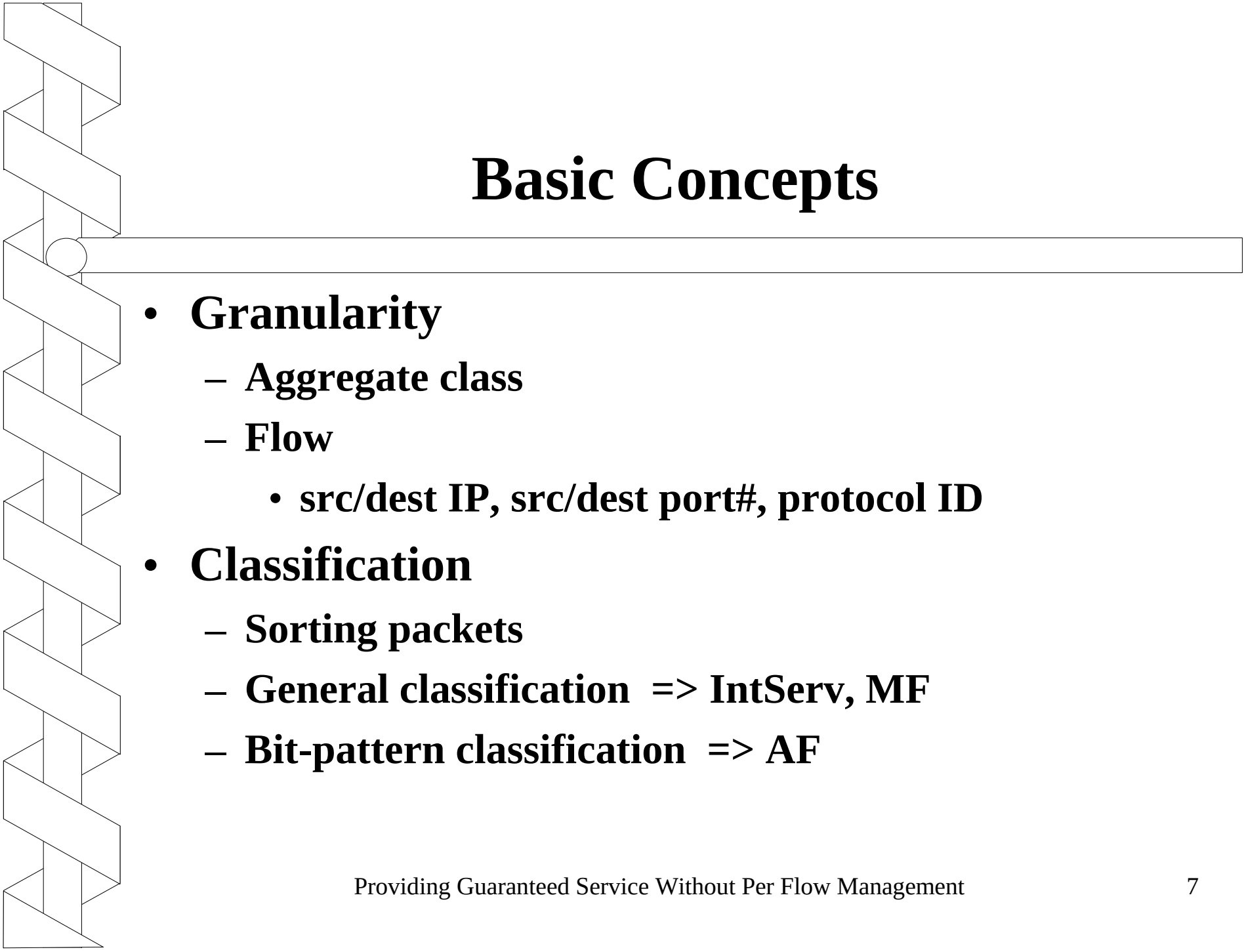
Internet QoS (2)

- **Current status**
  - **IP: best effort**
  - **TCP: reliable, sequential**

Advocate	Opponent
1. Diverse requirements	1. Provision: enough bandwidth
2. ISP: Better service, higher price	2. Applications adapt
3. Maximize utility	3. Complexity vs. benefit



1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**



Basic Concepts

- **Granularity**
  - **Aggregate class**
  - **Flow**
    - **src/dest IP, src/dest port#, protocol ID**
- **Classification**
  - **Sorting packets**
  - **General classification => IntServ, MF**
  - **Bit-pattern classification => AF**

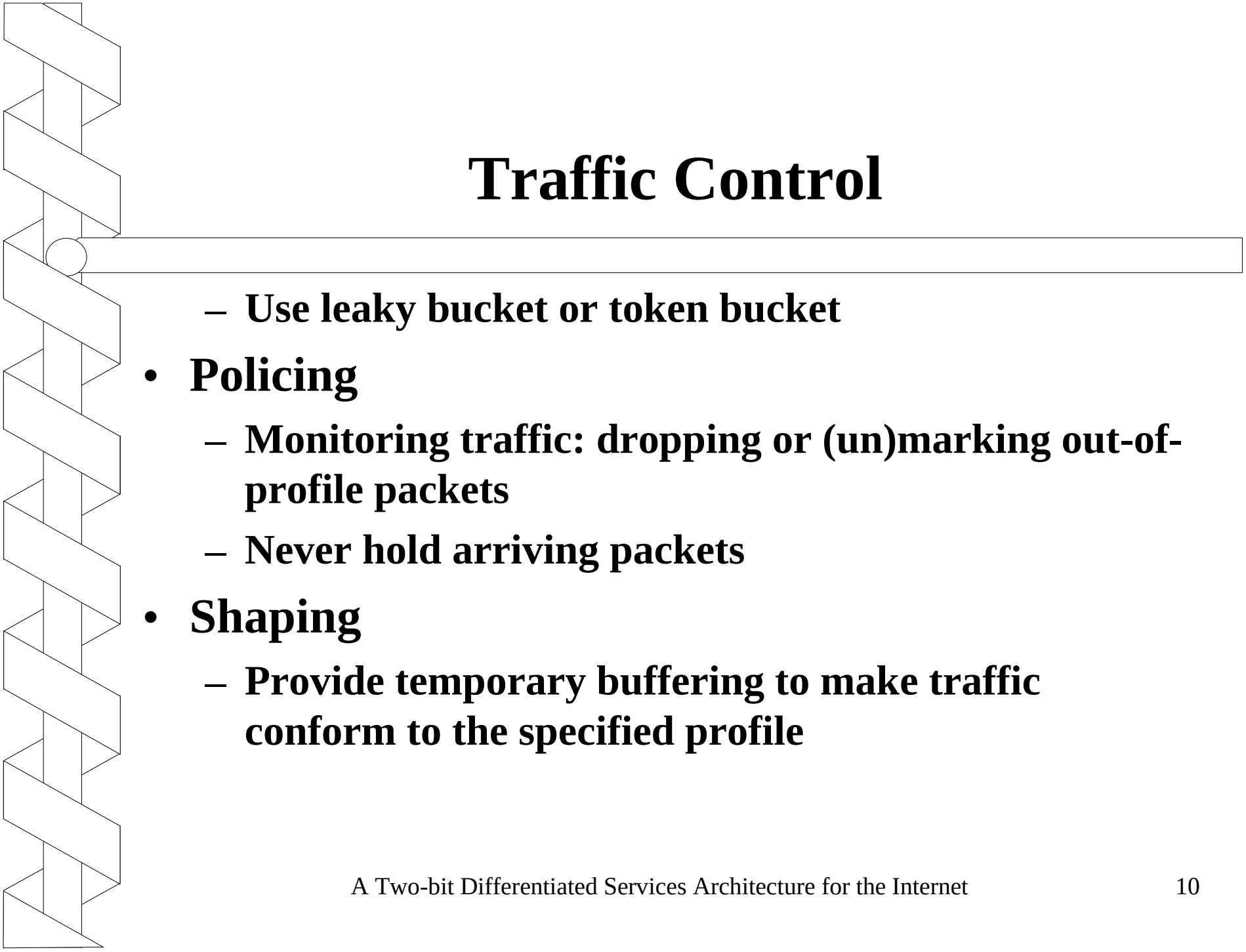
Specification

Traffic	Service
Traffic Profile Temporal properties **Tspec** Token bucket: token rate [r], bucket depth [b] Peak rate: [p] Min policed unit: [m] Max packet size: [M]	**Rspec** Per-flow based **SLA** (Service Level Agreement) Contract

Admission Control

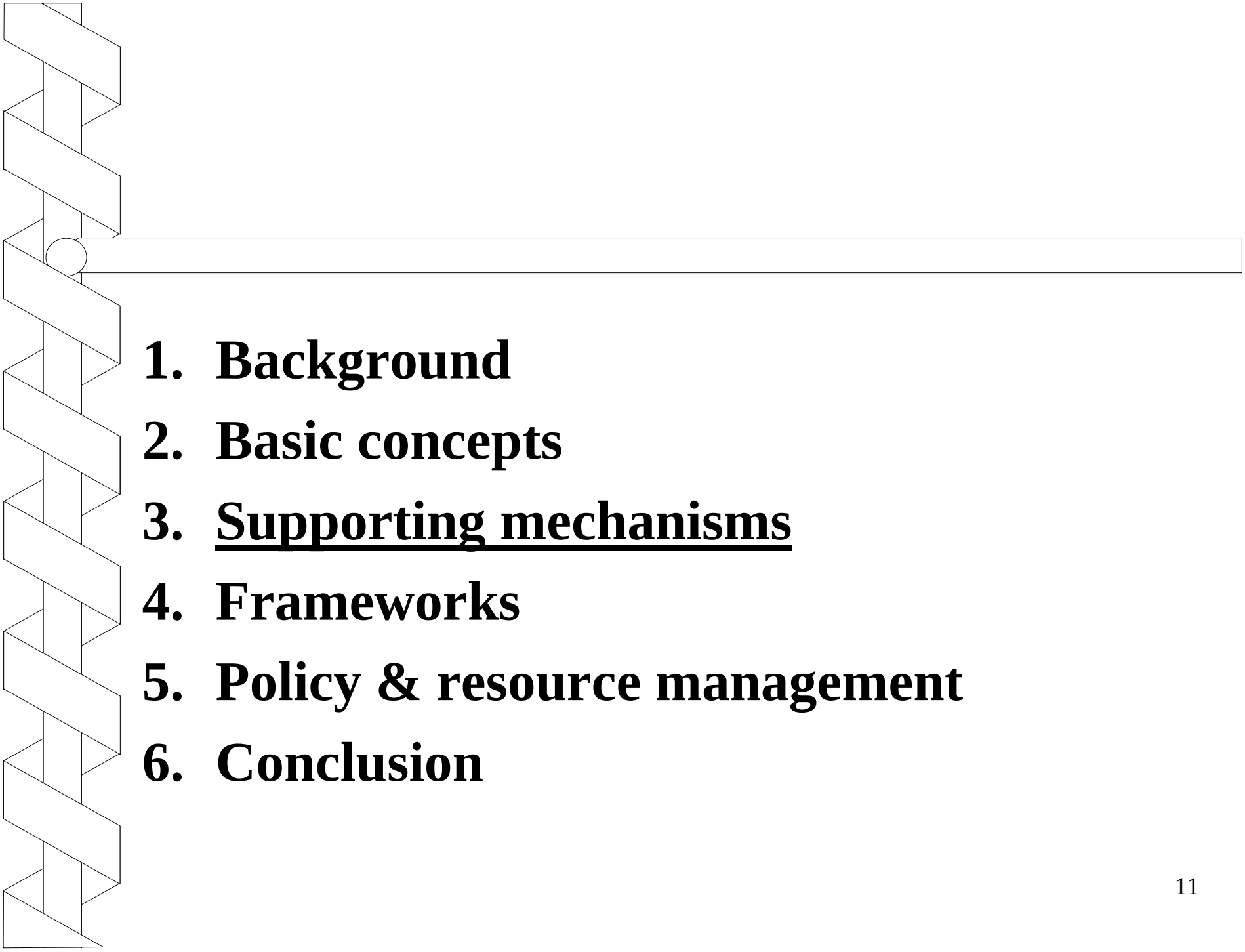
- Control resource allocation
- Decide whether to admit a new traffic stream

	Deterministic	Statistic	Measurement-based
QoS violation	No	Small probability	Occasional
Resource utilization	Low for bursty flows	high	high

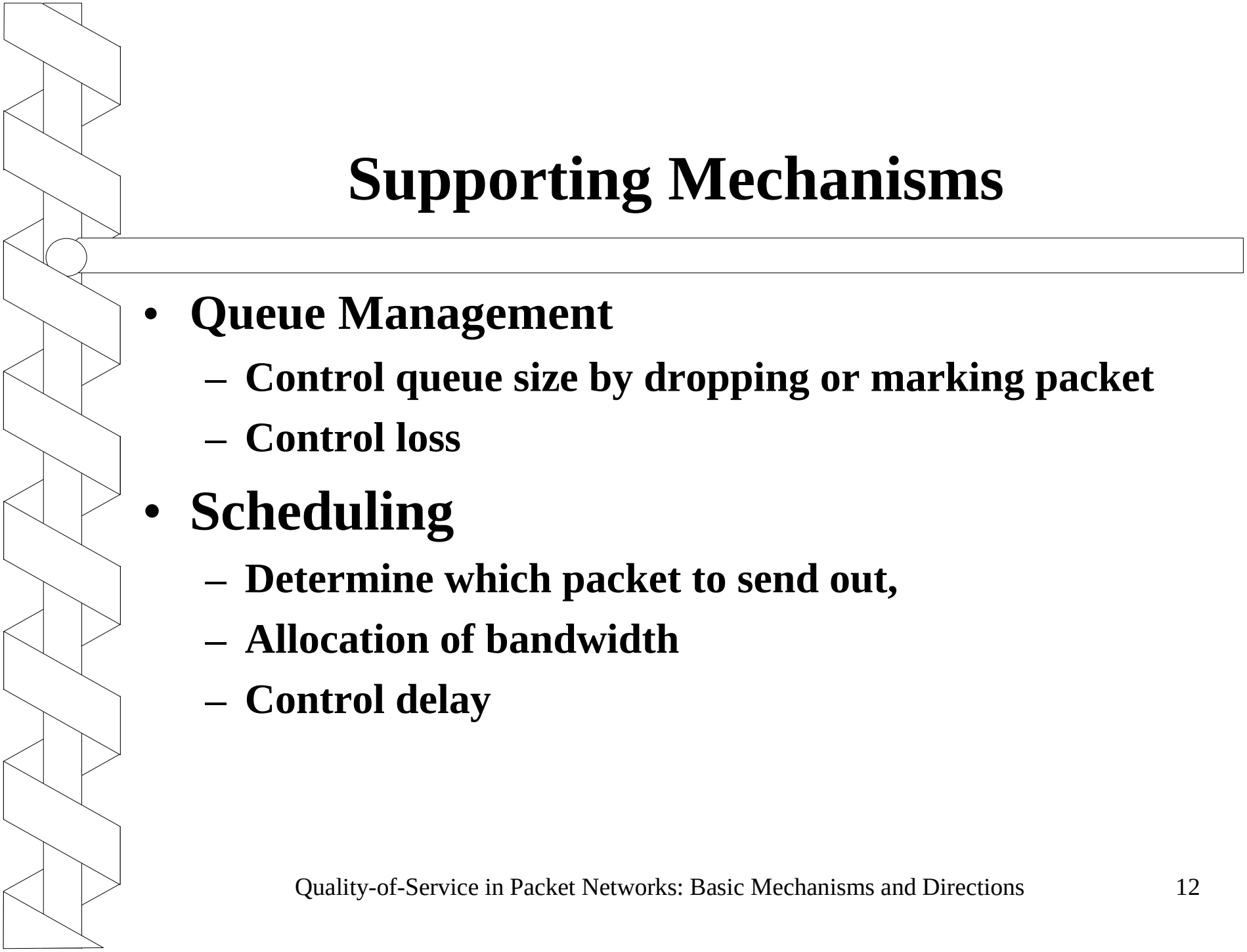


Traffic Control

  - Use leaky bucket or token bucket
- Policing
  - Monitoring traffic: dropping or (un)marking out-of-profile packets
  - Never hold arriving packets
- Shaping
  - Provide temporary buffering to make traffic conform to the specified profile



1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**

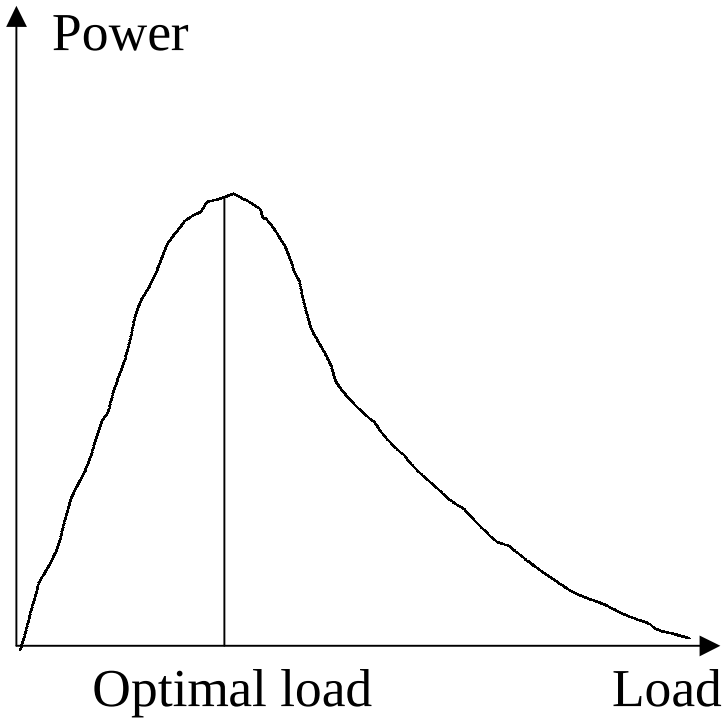


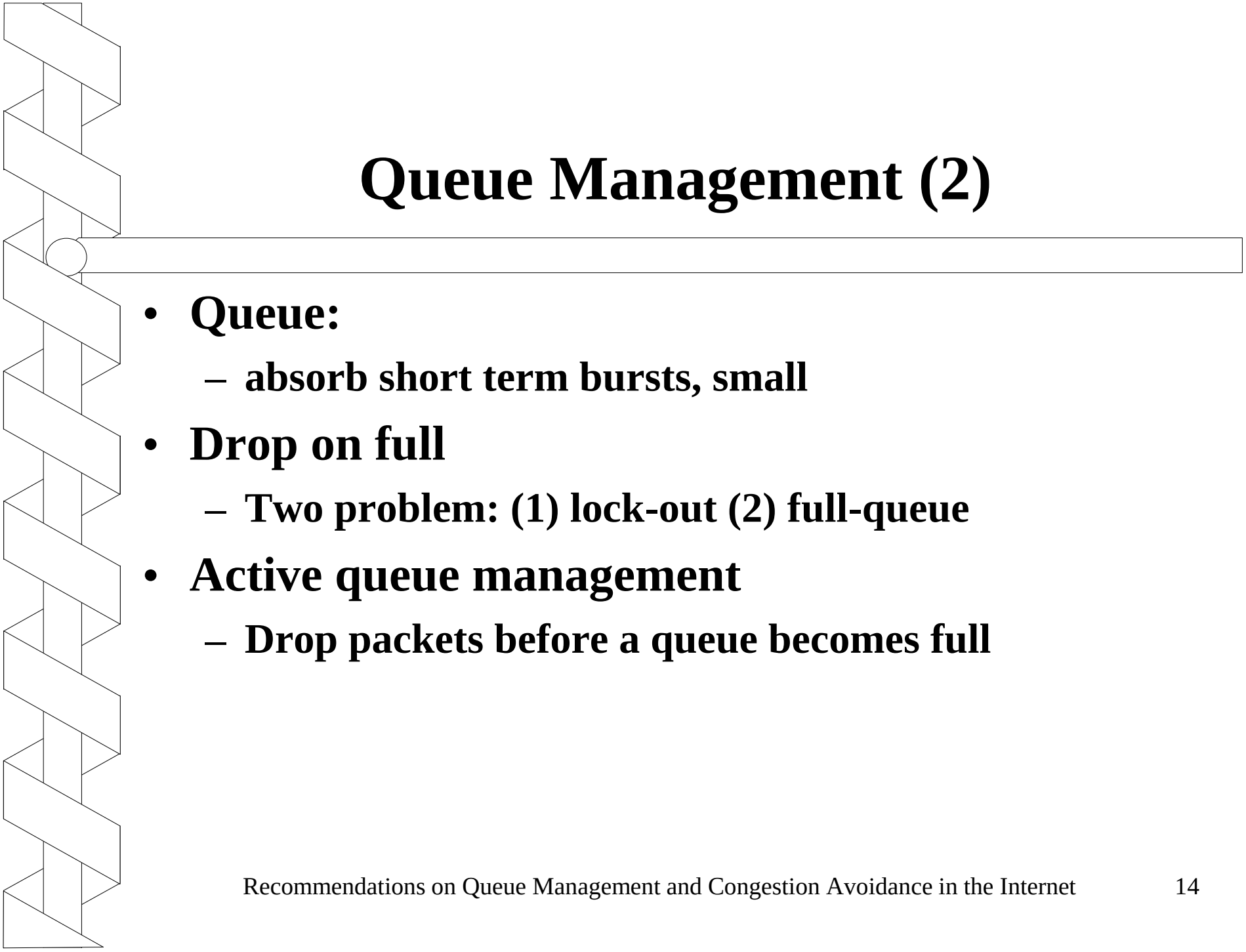
Supporting Mechanisms

- **Queue Management**
  - **Control queue size by dropping or marking packet**
  - **Control loss**
- **Scheduling**
  - **Determine which packet to send out,**
  - **Allocation of bandwidth**
  - **Control delay**

Queue Management

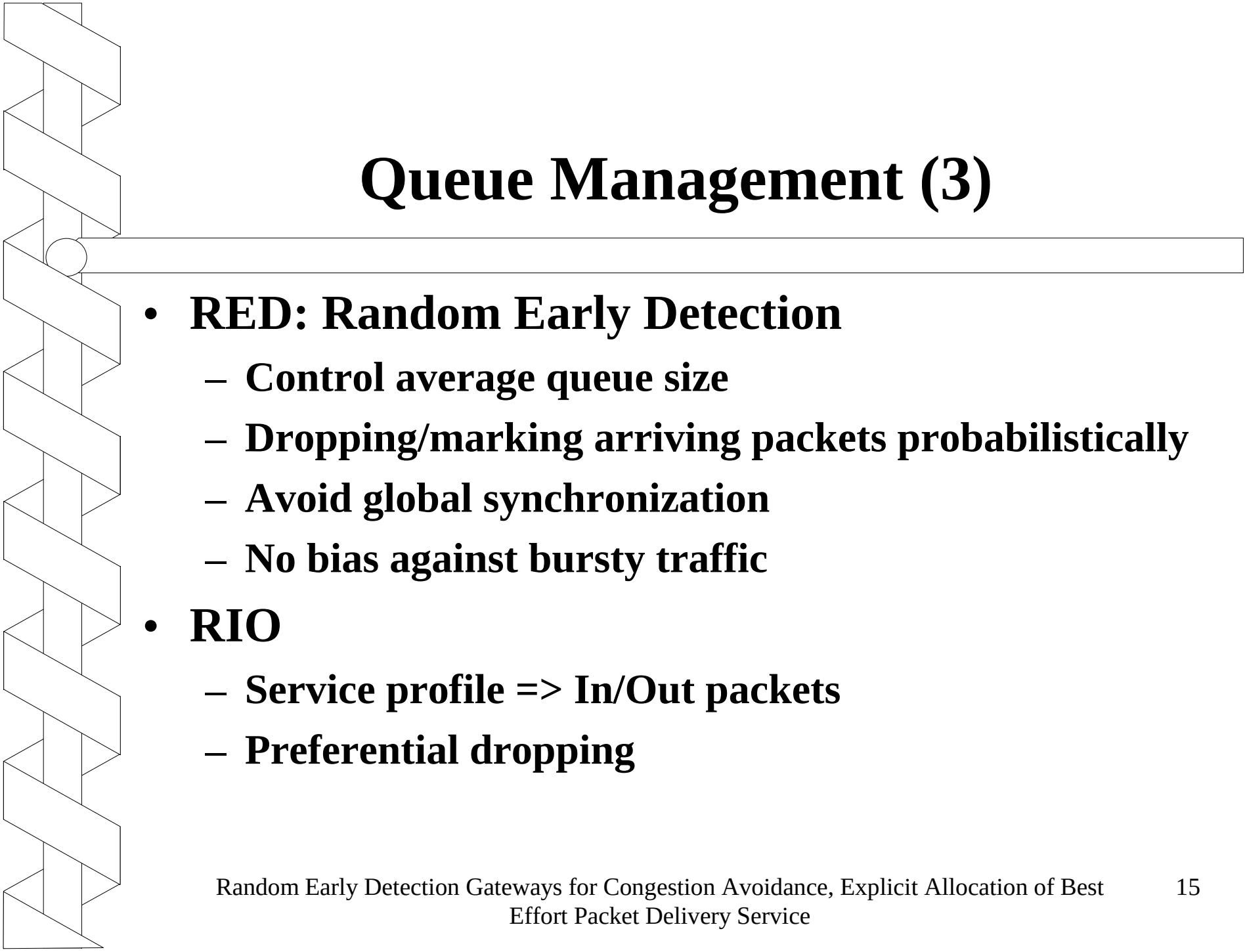
- **Loss**
  - **Damaged (<<1%)**
  - **Congestion**
- **Congestion control**
  - **End-point**
  - **Router**
  - **Goal: high throughput**
    low delay
    power=throughput/delay





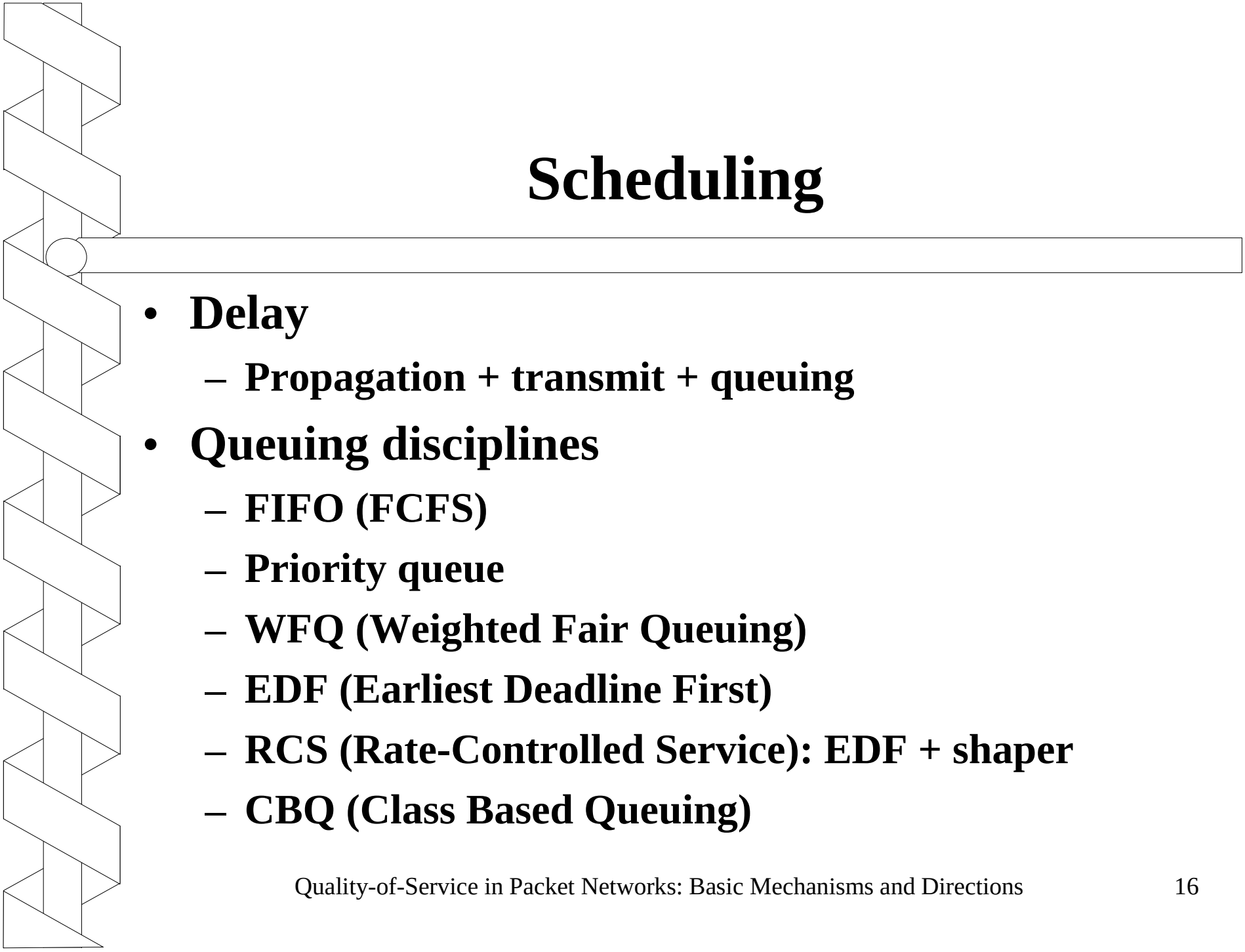
Queue Management (2)

- **Queue:**
  - **absorb short term bursts, small**
- **Drop on full**
  - **Two problem: (1) lock-out (2) full-queue**
- **Active queue management**
  - **Drop packets before a queue becomes full**



Queue Management (3)

- **RED: Random Early Detection**
  - **Control average queue size**
  - **Dropping/marking arriving packets probabilistically**
  - **Avoid global synchronization**
  - **No bias against bursty traffic**
- **RIO**
  - **Service profile => In/Out packets**
  - **Preferential dropping**

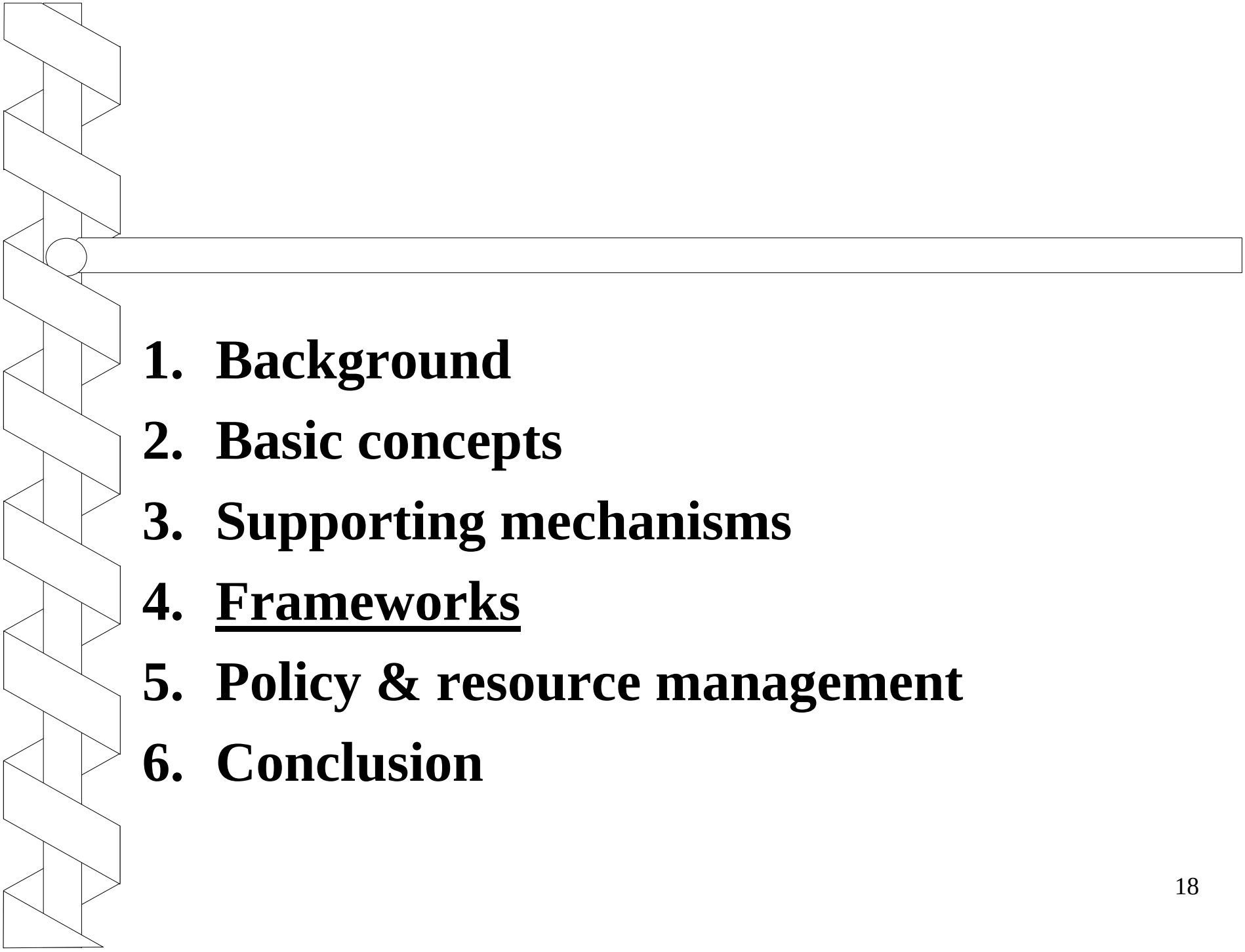


Scheduling

- **Delay**
  - **Propagation + transmit + queuing**
- **Queuing disciplines**
  - **FIFO (FCFS)**
  - **Priority queue**
  - **WFQ (Weighted Fair Queuing)**
  - **EDF (Earliest Deadline First)**
  - **RCS (Rate-Controlled Service): EDF + shaper**
  - **CBQ (Class Based Queuing)**

Scheduling (2)

- **Link sharing**
  - **Share aggregated bandwidth in a controlled way under overload**
  1. **multi-entity**
  2. **multi-protocol**
  3. **multi-service**
- **Hierarchical link sharing:**
  - **GPS (Generalized Processor Sharing)**
  - **A theoretic reference model**



1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**

IntServ

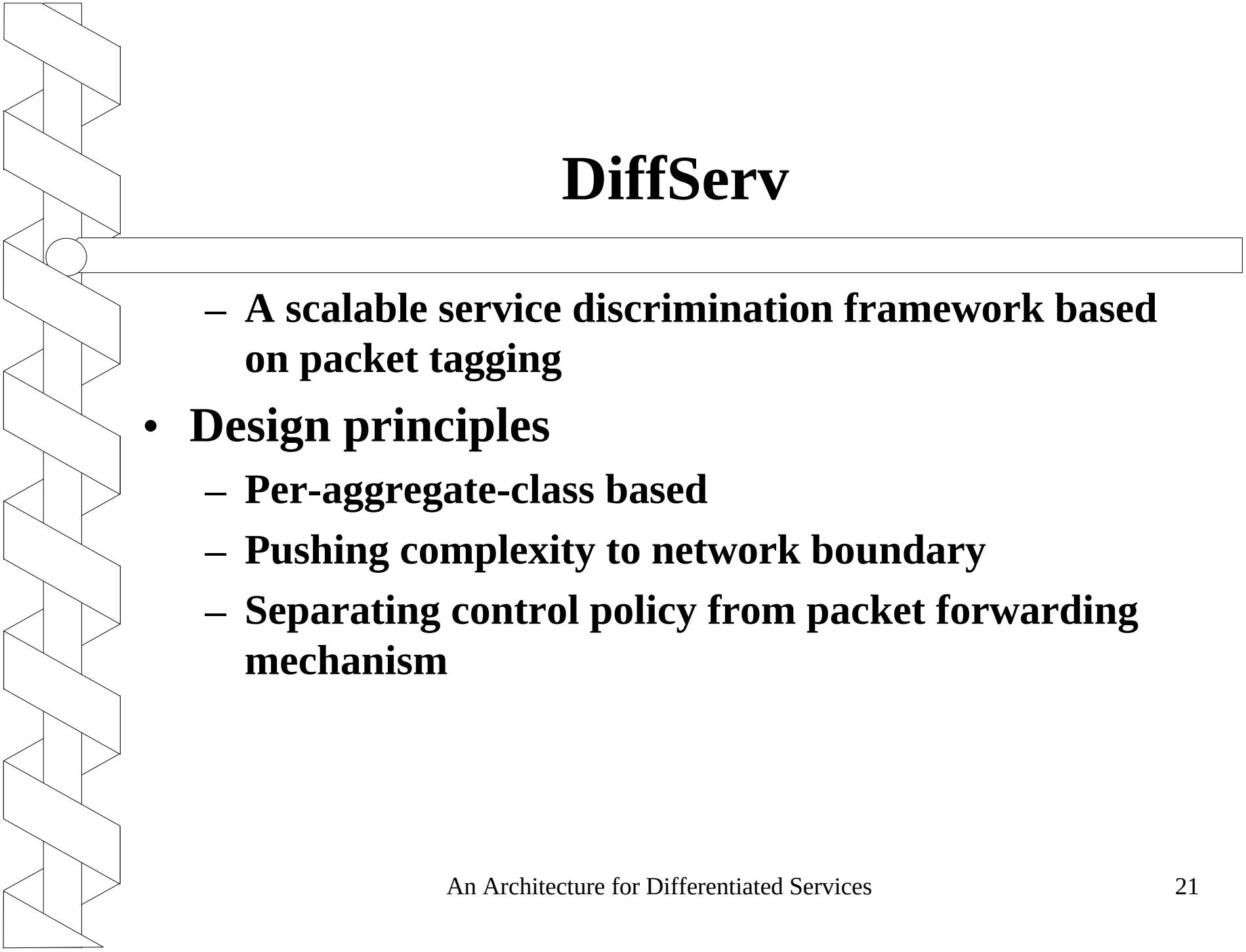
- **Per-flow based QoS framework with dynamic resource reservation**
  - **Control path: RSVP, admission control**
  - **Data path: classification, scheduling**
- **RSVP**
  - **Signaling protocol: path setup, resource reservation**
  - **Receiver initiation**
  - **Soft state for robust**

IntServ (2)

- **Service models**

	Guaranteed	Controlled-load
Goal	Control max queuing delay	Closely equivalent to unloaded best effort service
Deployment	ubiquitous	incremental

- **Scalability problem**
  - **flow state @ router**



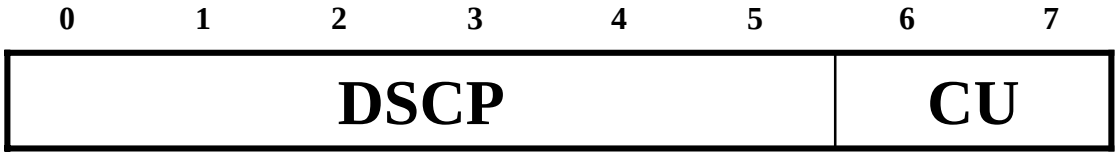
DiffServ

- A scalable service discrimination framework based on packet tagging

- **Design principles**
  - Per-aggregate-class based
  - Pushing complexity to network boundary
  - Separating control policy from packet forwarding mechanism

DiffServ (2)

- **DS field**
  - **redefine TOS field in IPv4 header**



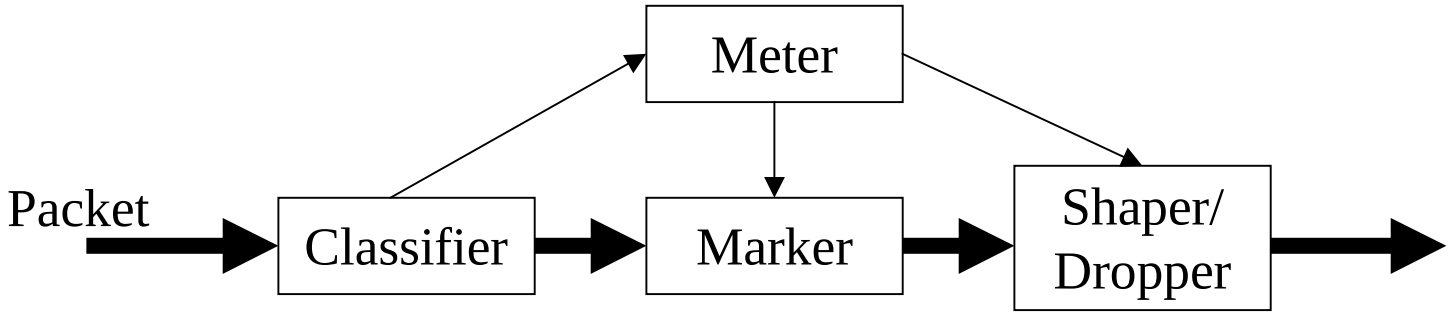
DSCP: Differentiated Services Codepoint

CU: Currently Unused

- **PHB**
  - **Per-hop behavior**

DiffServ (3)

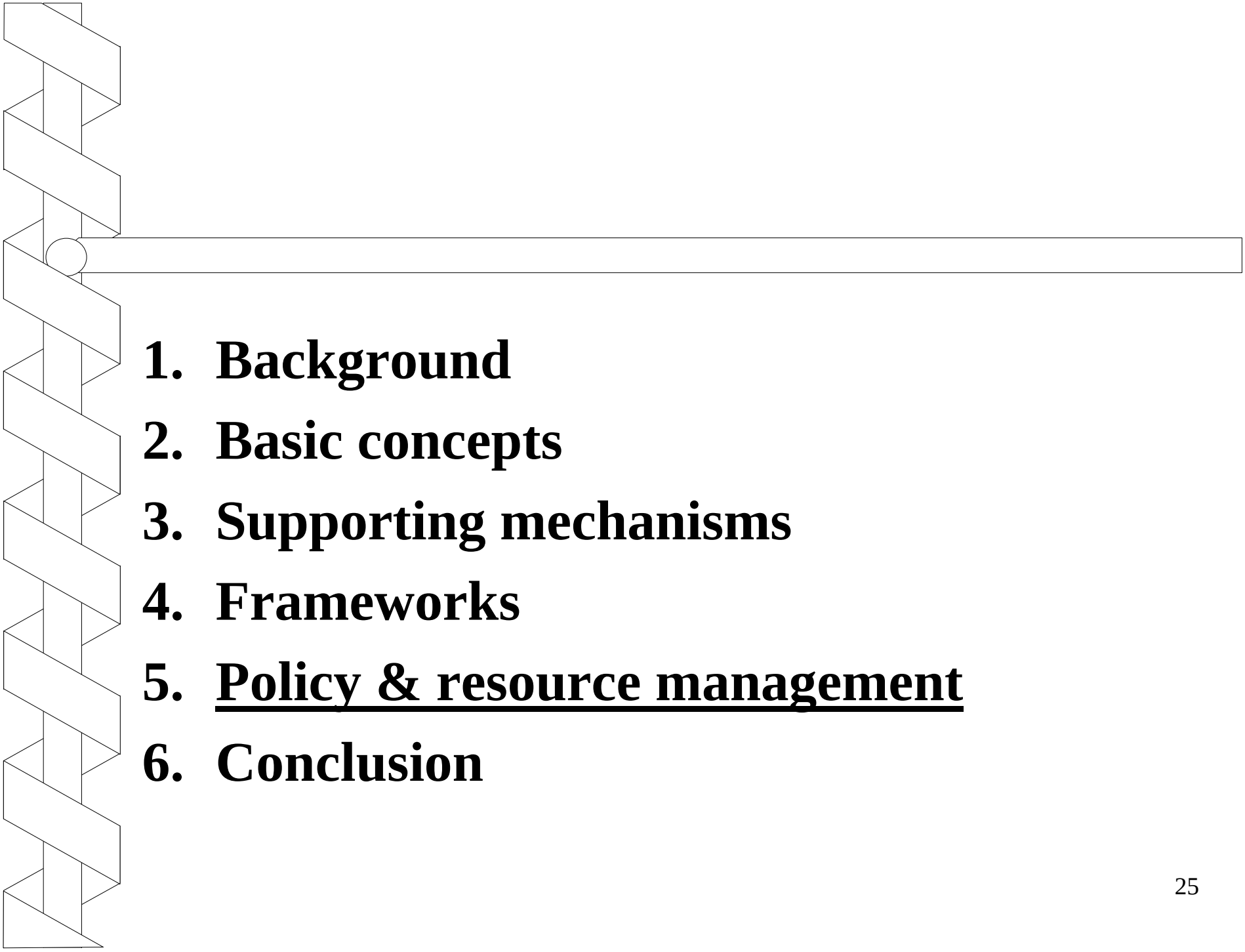
- **Network boundary**
  - **Edge routers, leaf routers, hosts**
  - **Finer granularity: classification, conditioning**
- **Core router**
  - **simple PHB: fast & scalable**



DiffServ (4)

- **Service Models**

Premium	Assured
Guaranteed peak rate	Statistical provisioning
Little queuing delay	In: unlikely dropped
Smoothing bursts	Out: preferential dropping
Virtual-leased line (absolute)	Olympic service (relative)



1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**

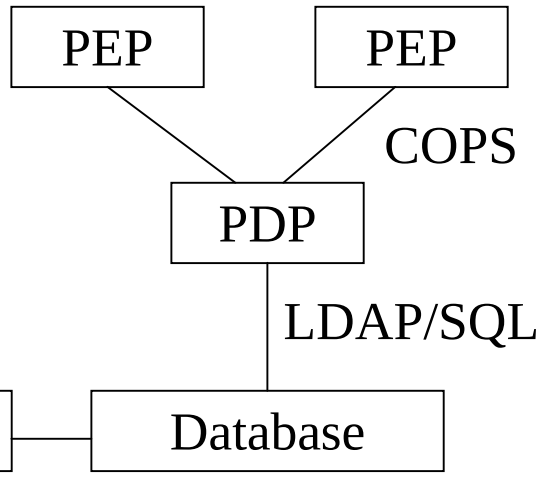
Policy

- **regulation of access to network resources & services**
- **<u>Policy infrastructure</u>: administrative intentions ⇨ differential packet treatment**

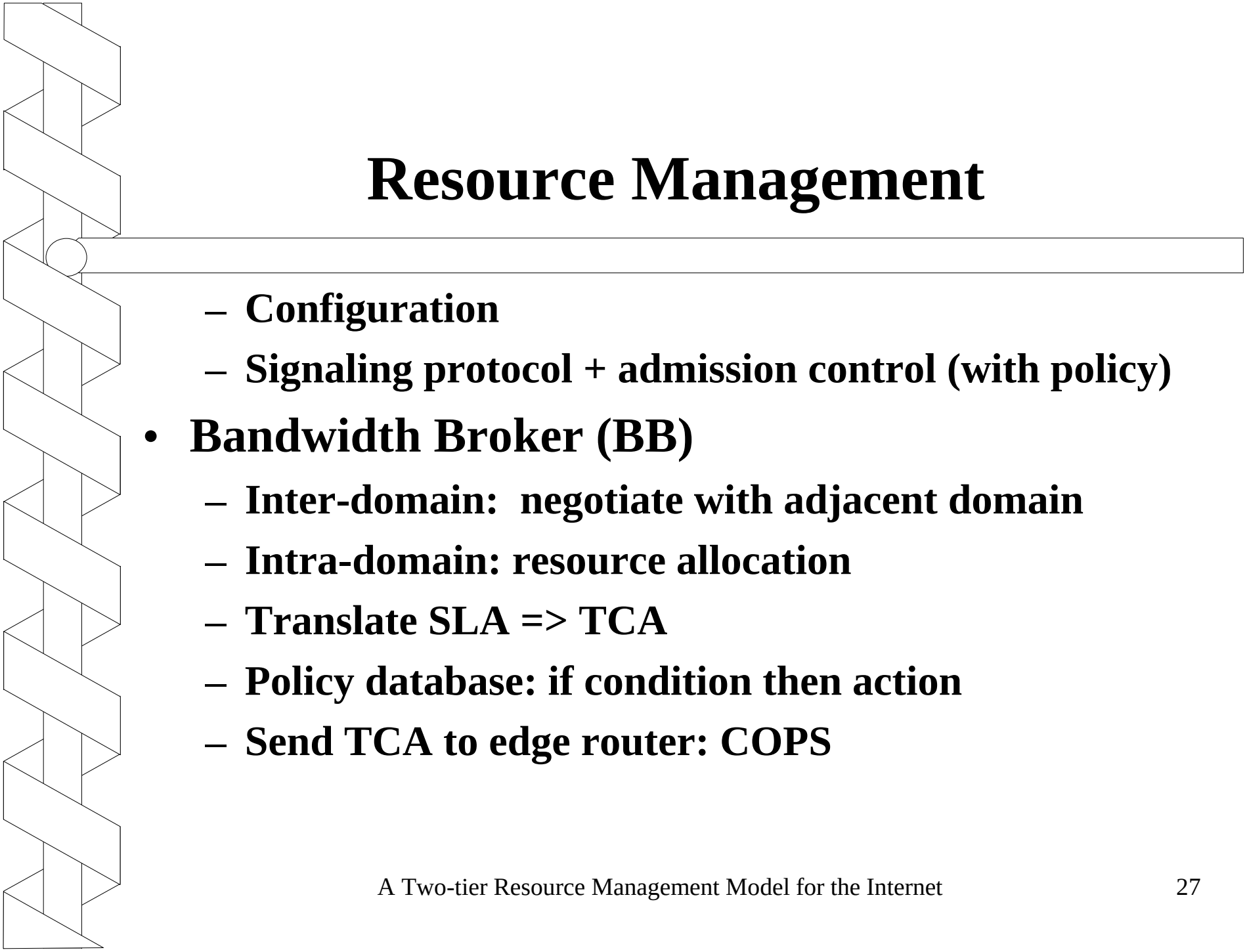
PEP: Policy Enforcement Point

PDP: Policy Decision Point

COPS: Common Open Policy Service



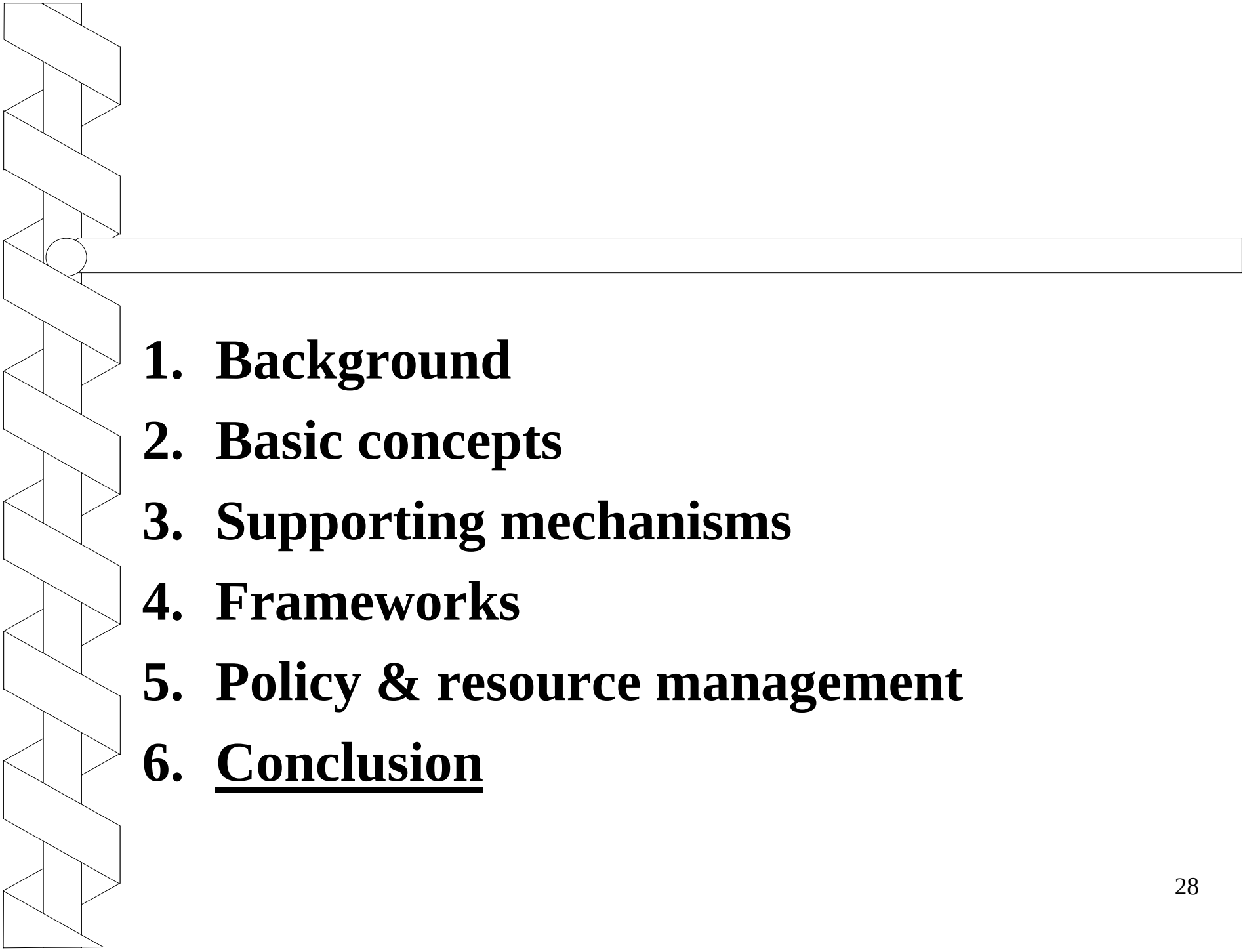
Policy Domain



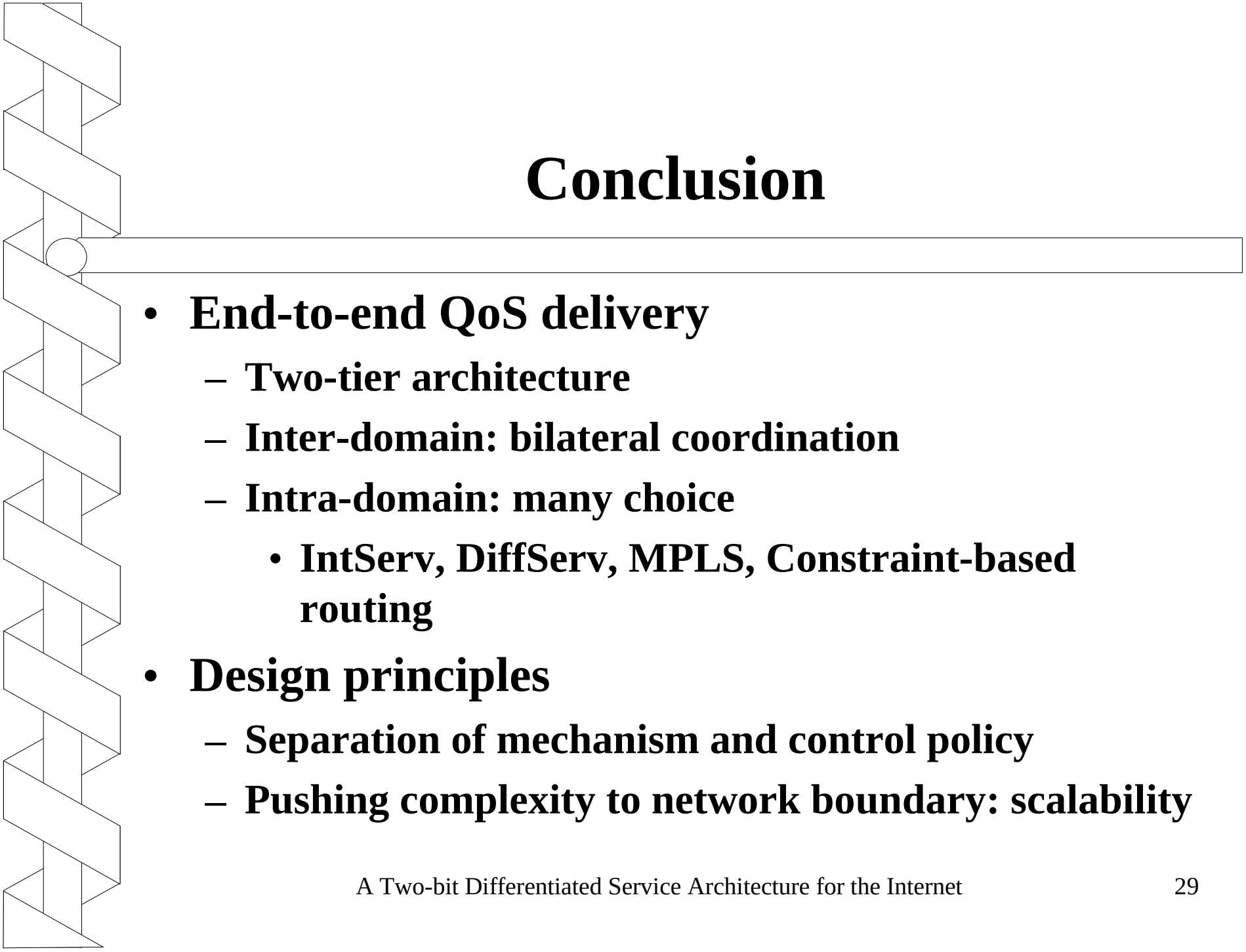
Resource Management

- **Configuration**
- **Signaling protocol + admission control (with policy)**

- **Bandwidth Broker (BB)**
  - **Inter-domain: negotiate with adjacent domain**
  - **Intra-domain: resource allocation**
  - **Translate SLA => TCA**
  - **Policy database: if condition then action**
  - **Send TCA to edge router: COPS**



1. **Background**
2. **Basic concepts**
3. **Supporting mechanisms**
4. **Frameworks**
5. **Policy & resource management**
6. **Conclusion**



Conclusion

- **End-to-end QoS delivery**
  - **Two-tier architecture**
  - **Inter-domain: bilateral coordination**
  - **Intra-domain: many choice**
    - **IntServ, DiffServ, MPLS, Constraint-based routing**
- **Design principles**
  - **Separation of mechanism and control policy**
  - **Pushing complexity to network boundary: scalability**