NETWORK

DESIGN & ANALYSIS

04 FLOW ANALYSIS

Contents

4.1 Background

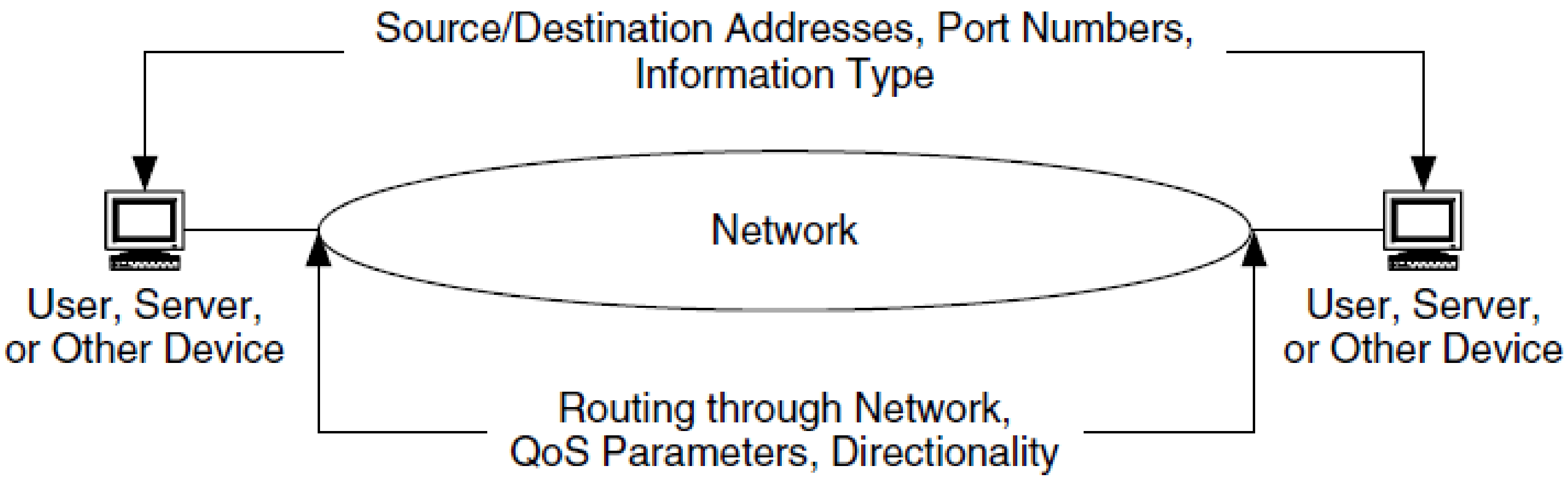
- Further analyze of the user, application, device, and network requirements based on their end-to-end characteristics.

- This chapter introduces flows and flow concepts, data sources and sinks, and flow models which will help us to identify, size, and describe flows.

4.2 Flows

- *Flows* (also known as *traffic flows* or *data flows*) are sets of network traffic (application, protocol, and control information) that have common attributes, such as source/destination address, type of information, directionality, or other end-to-end information.

4.2 Flows


Source/Destination Addresses, Port Numbers,
Information Type
Network
User, Server,
or Other Device
User, Server,
or Other Device
Routing through Network,
QoS Parameters, Directionality

4.2 Flows

Flow Characteristics	
Performance Requirements	Capacity (e.g., Bandwidth)
	Delay (e.g., Latency)
	Reliability (e.g., Availability)
	Quality of Service Levels
Importance/ Priority Levels	Business/Enterprise/Provider
	Political
Other	Directionality
	Common Sets of Users, Applications, Devices
	Scheduling (e.g., Time-of-Day)
	Protocols Used
	Addresses/Ports
	Security/Privacy Requirements

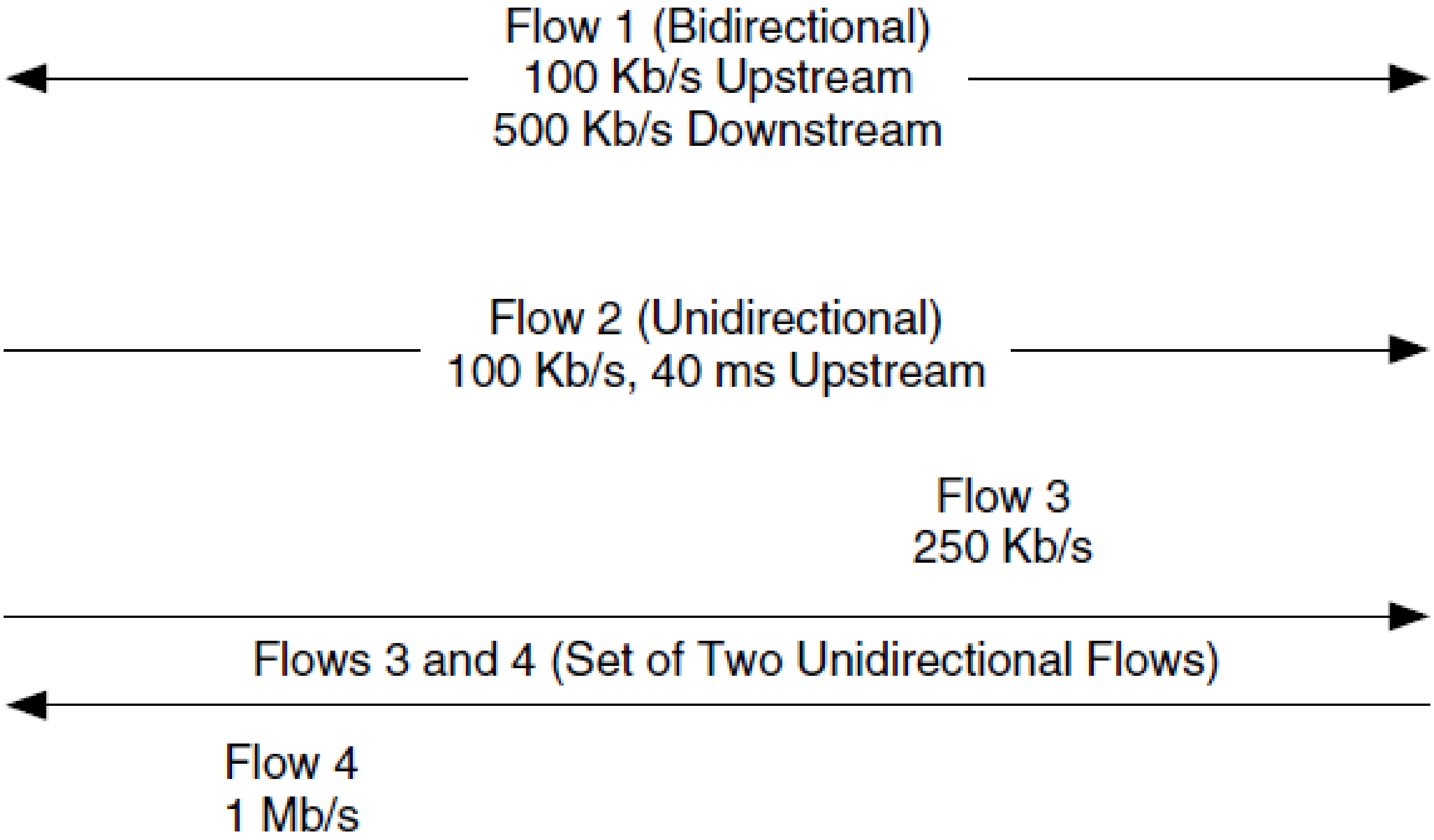
4.2 Flows

- Flow analysis provides an end-to-end perspective on requirements and shows where requirements combine and interact.
- It can be represented as either a single, double-sided arrow with one or two sets of performance requirements, or as two separate flows, each with its own set of requirements.

4.2 Flows

- Flows provide a different perspective on traffic movement in networks: they have logical as well as physical components; and they allow traffic to be coupled with users, applications, or devices.
- Two types of flows: individual and composite.

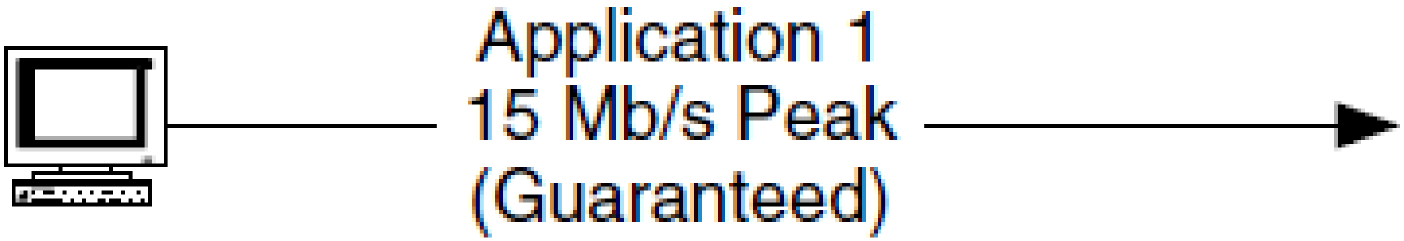
4.2 Flows


Flow 1 (Bidirectional)
100 Kb/s Upstream
500 Kb/s Downstream
Flow 2 (Unidirectional)
100 Kb/s, 40 ms Upstream
Flow 3
250 Kb/s
Flows 3 and 4 (Set of Two Unidirectional Flows)
Flow 4
1 Mb/s

4.2 Flows

4.2.1 Individual & Composite Flows

– An *individual flow* is the flow for a single session of an application.

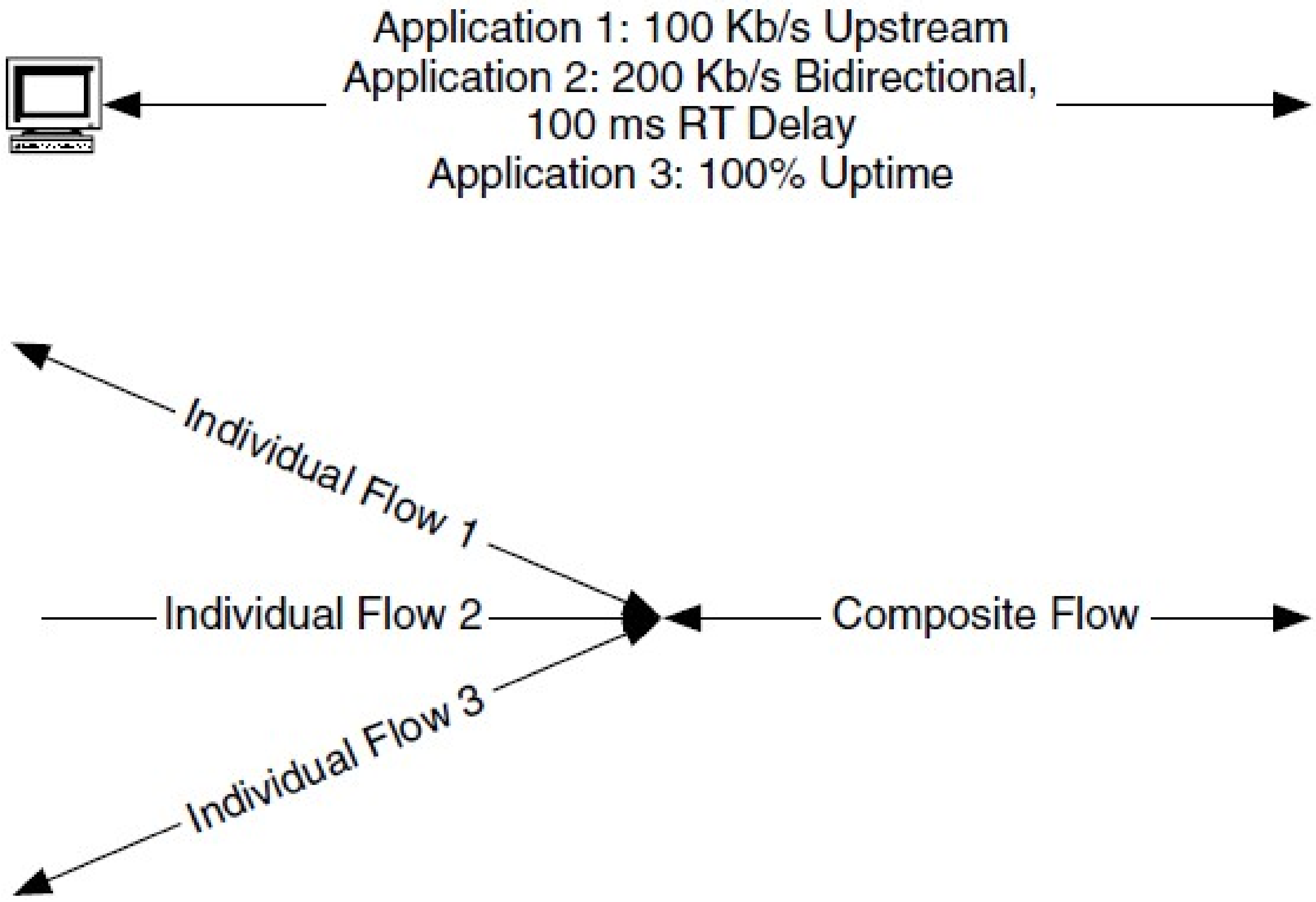


4.2 Flows

– A *composite flow* is a combination of requirements from multiple applications, or of individual flows, that share a common link, path, or network.

– Most flows in a network are composites.

4.2 Flows



Application 1: 100 Kb/s Upstream
Application 2: 200 Kb/s Bidirectional,
100 ms RT Delay
Application 3: 100% Uptime
Individual Flow 1
Individual Flow 2
Individual Flow 3
Composite Flow

4.2 Flows

4.2.2 Critical Flows

- Some flows can be considered more important than others, in that they are higher in performance or have strict requirements.
- Those are called critical flows.
- Flows that require high, predictable, and/or guaranteed performance are the ones that drive the architecture and design from a service (capacity, delay, and RMA) perspective.

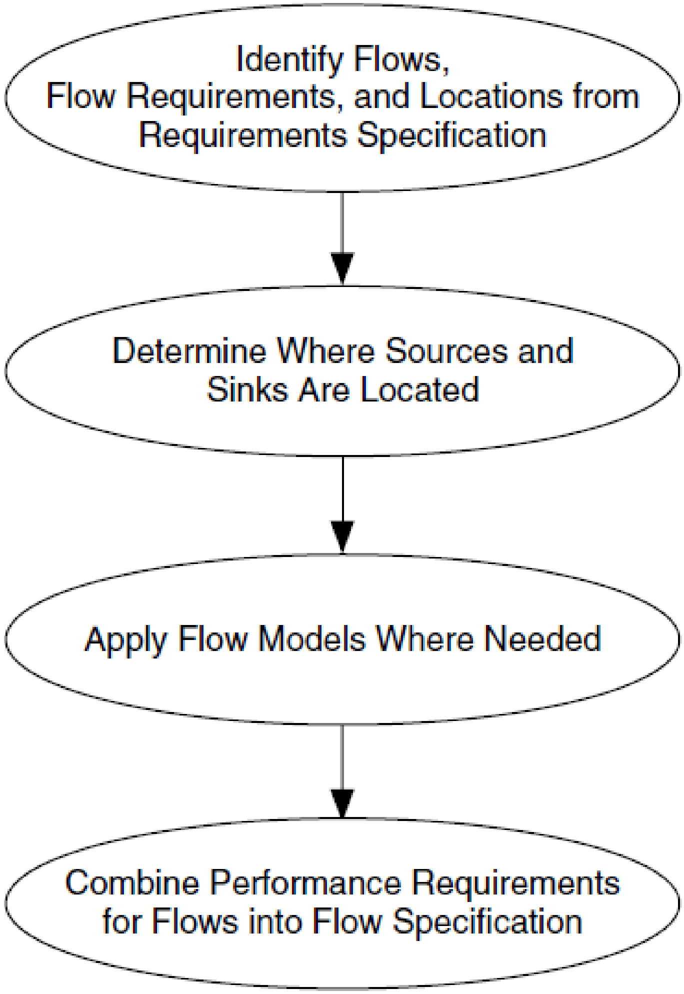
4.3 Flows Identification

- Flows can usually be identified and developed from information
  - in the requirements specification: user, application, device, and network requirements;
  - user and application behavior (usage patterns, models); user, application, and device location information; and
  - performance requirements.

4.3 Flows Identification

- Flows are determined based on the requirements and locations of the applications and devices that generate (source) or terminate (sink) each traffic flow.
- Flows identification consists of identifying one or more applications and/or devices that will generate and/or terminate traffic flows.

4.3 Flows Identification



4.3 Flows Identification

- From an application perspective, some common approaches to identifying flows include:
  - Focusing on a particular application, application group, device, or function (e.g., videoconferencing or storage).
  - Developing a “profile” of common or selected applications that can be applied across a user population.

4.3 Flows Identification

– Choosing the top N (e.g., 3, 5, 10, etc.) applications to be applied across the entire network.

4.3 Flows Identification

4.3.1 Particular Application

- The idea here is to consider one or more applications that will likely drive the architecture and design.
- Namely, those that are high performance, mission-critical, rate-critical, real-time, interactive, predictable, and/or guaranteed.
- Choose what to focus on and select the relevant information from the requirements specification.

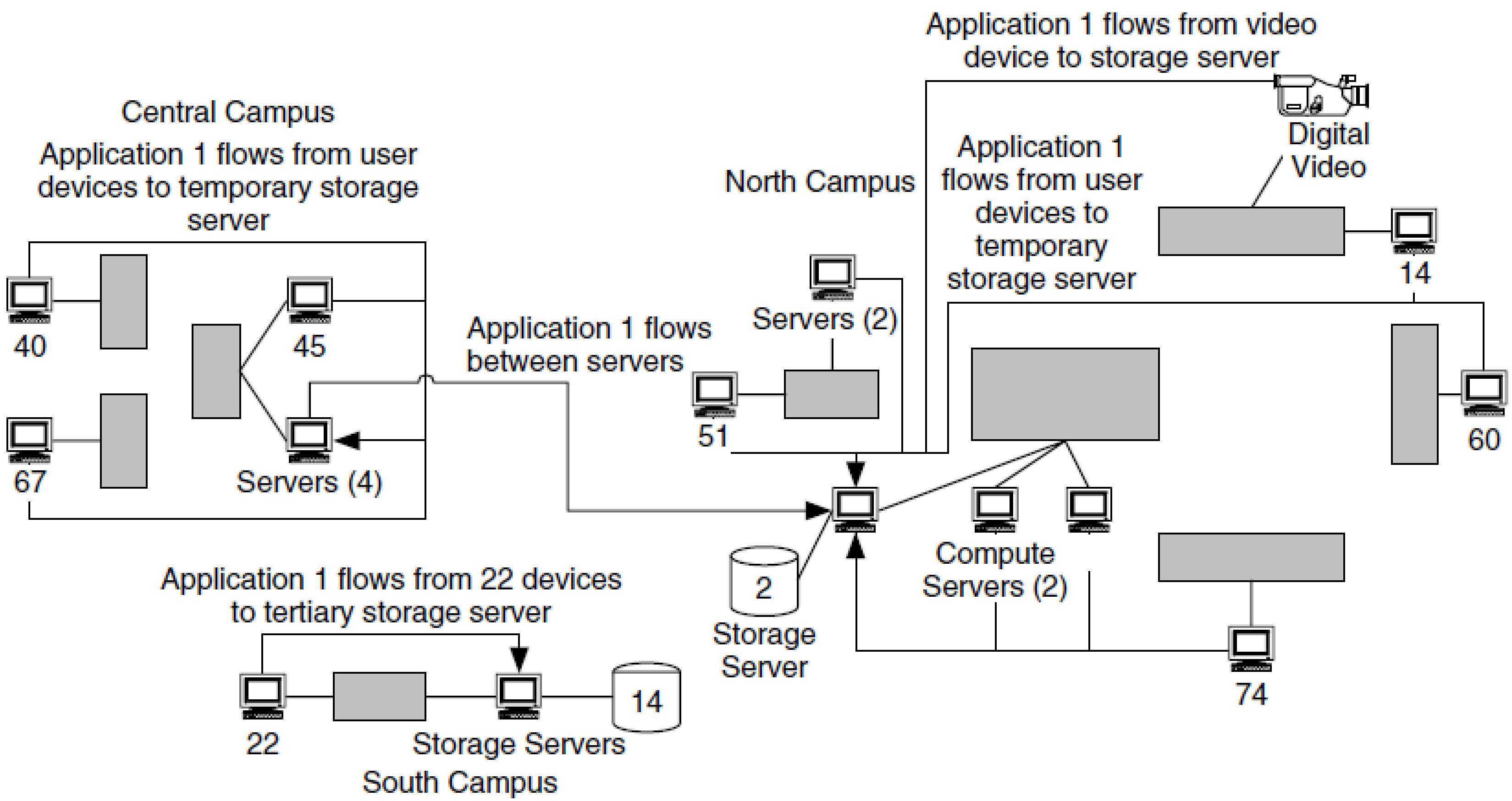
4.3 Flows Identification

- One example is Data Migration. From requirements specification, for a single session of each application:
  - Application 1: Staging data from user devices
  - Capacity 100 Kb/s; Delay Unknown; Reliability 100%
  - Application 1: Migrating data between servers
  - Capacity 500 Kb/s; Delay Unknown; Reliability 100%
  - Application 2: Migration to remote (tertiary) storage
  - Capacity 10Mb/s; Delay N/A; Reliability 100%

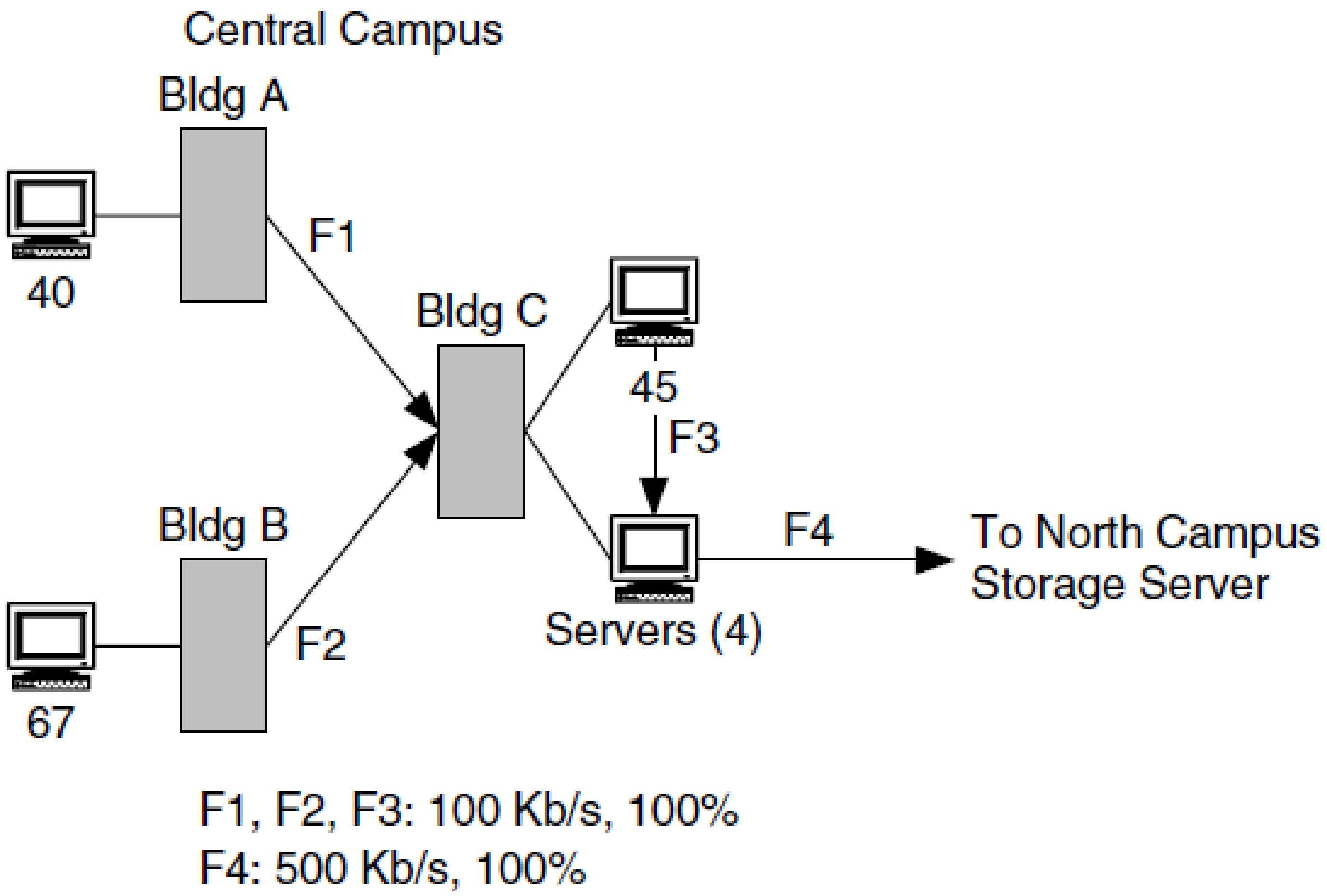
4.3 Flows Identification

- Location information can be used to map out where the users, applications and devices are.
- Once flows are mapped, performance information are applied to each flow.

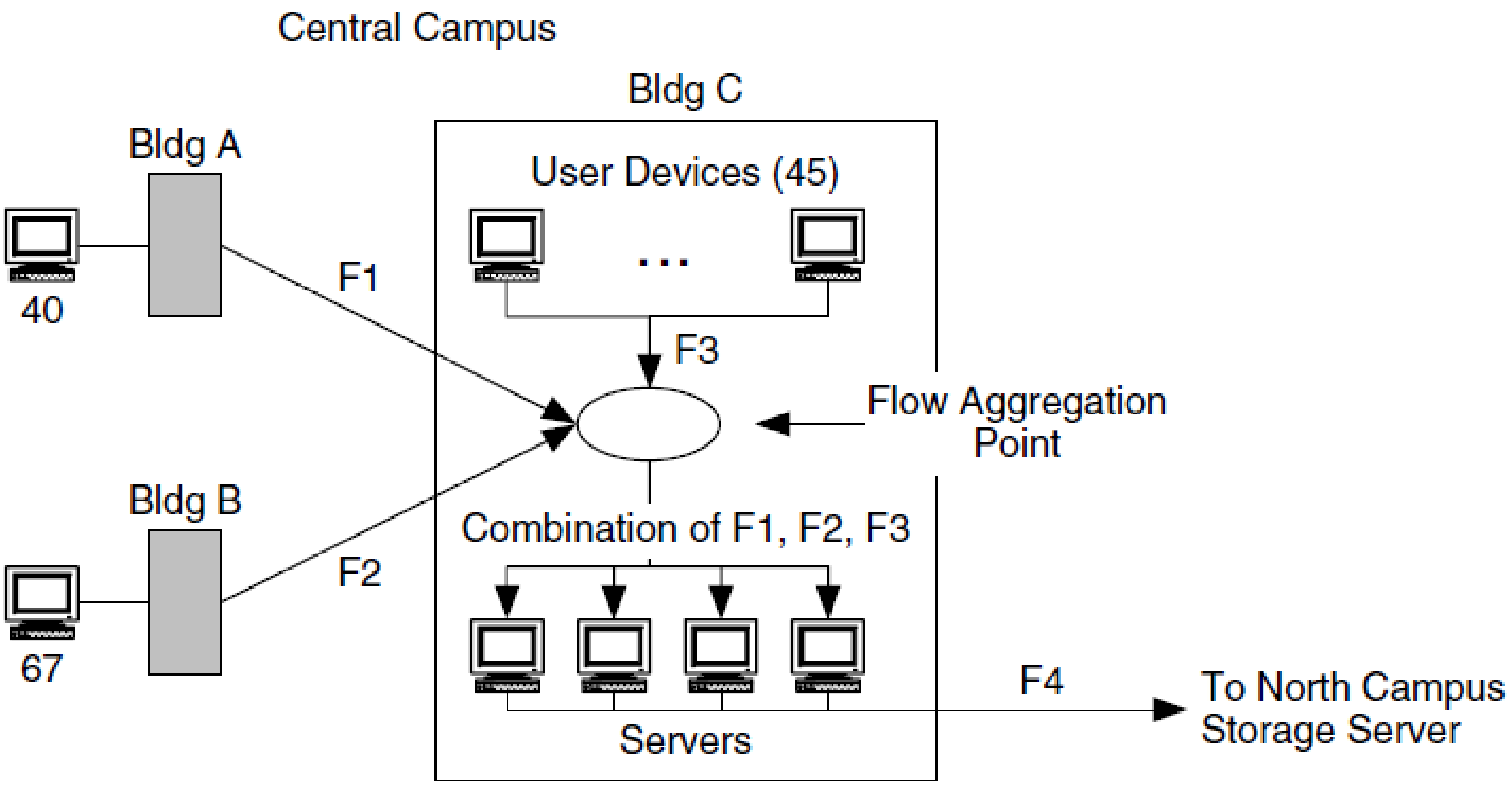
4.3 Flows Identification



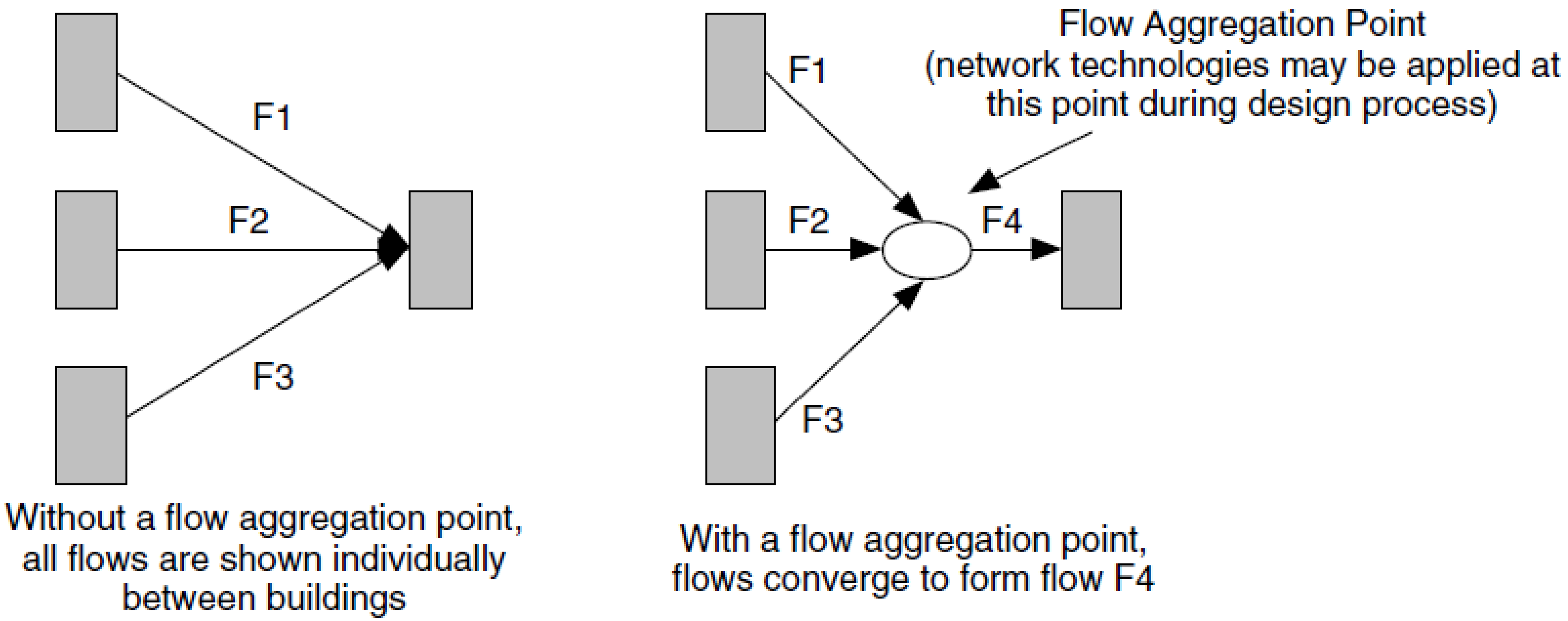
4.3 Flows Identification



4.3 Flows Identification



4.3 Flows Identification

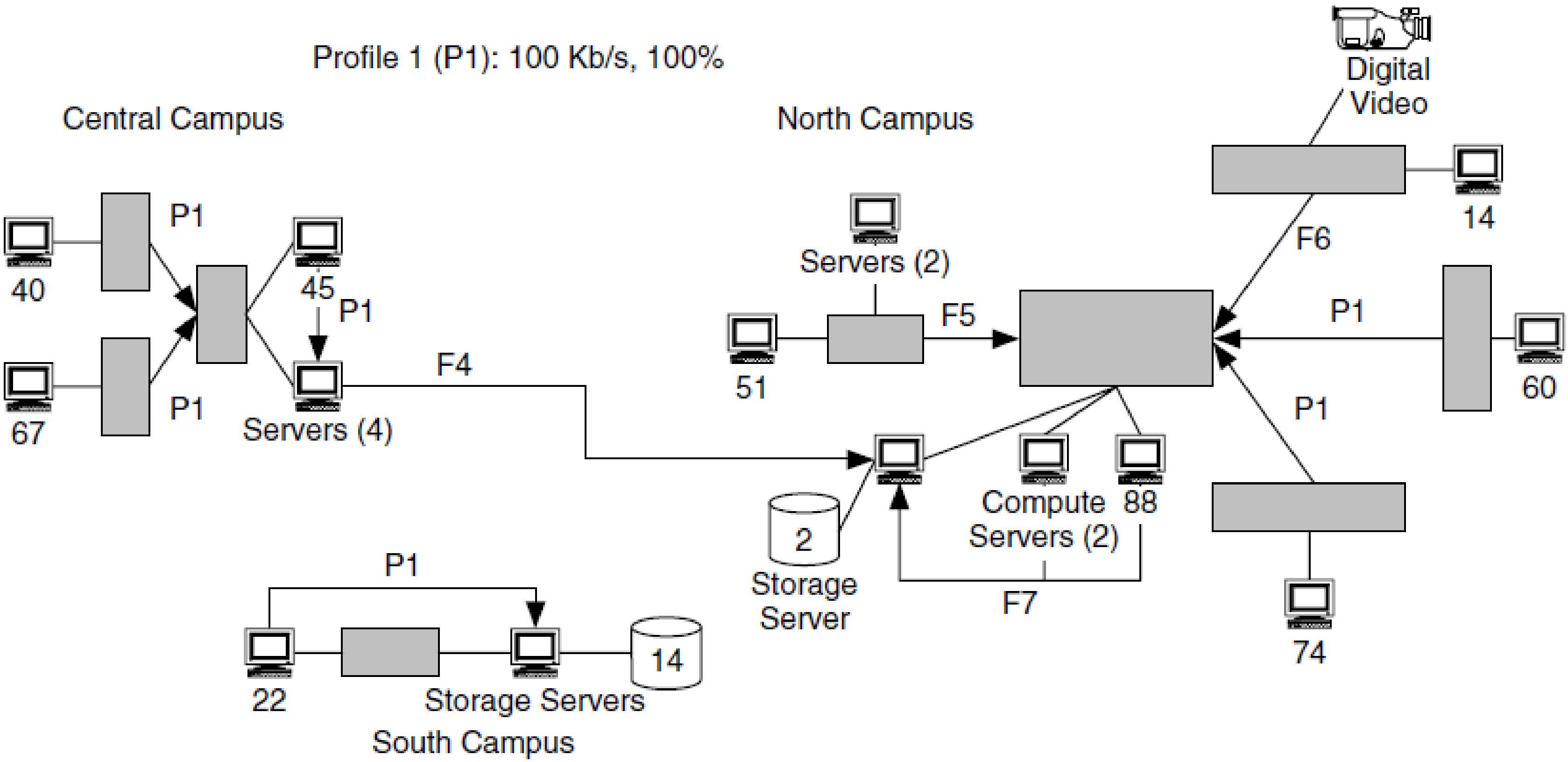


4.3 Flows Identification

4.3.2 Developing a Profile

- Sometimes a set of common applications apply to a group of users or to the entire set of users.
- When this is the case a profile or template can be developed for those applications, and each flow that fits the profile is identified with that profile's tag.

4.3 Flows Identification



4.3 Flows Identification

- In this figure, P1 is the tag associated with flows having the following performance requirements: capacity=100 Kb/s, reliability=100%.
- There are six flows in this diagram with those performance requirements, all of which are tagged as P1.
- Flow F4 has different performance requirements: capacity=500 kb/s, reliability=100%.

4.3 Flows Identification

- Flow 5 combines the performance requirements of 51 users (which, for Application 1, would be P1) with those of the two servers in that building.
- Flow F6 combines the performance requirements of 14 users (again, this would be P1) with that of the digital video device.
- F7, like F5, combines the performance requirements of users (88 in this building) with those of the two compute servers.

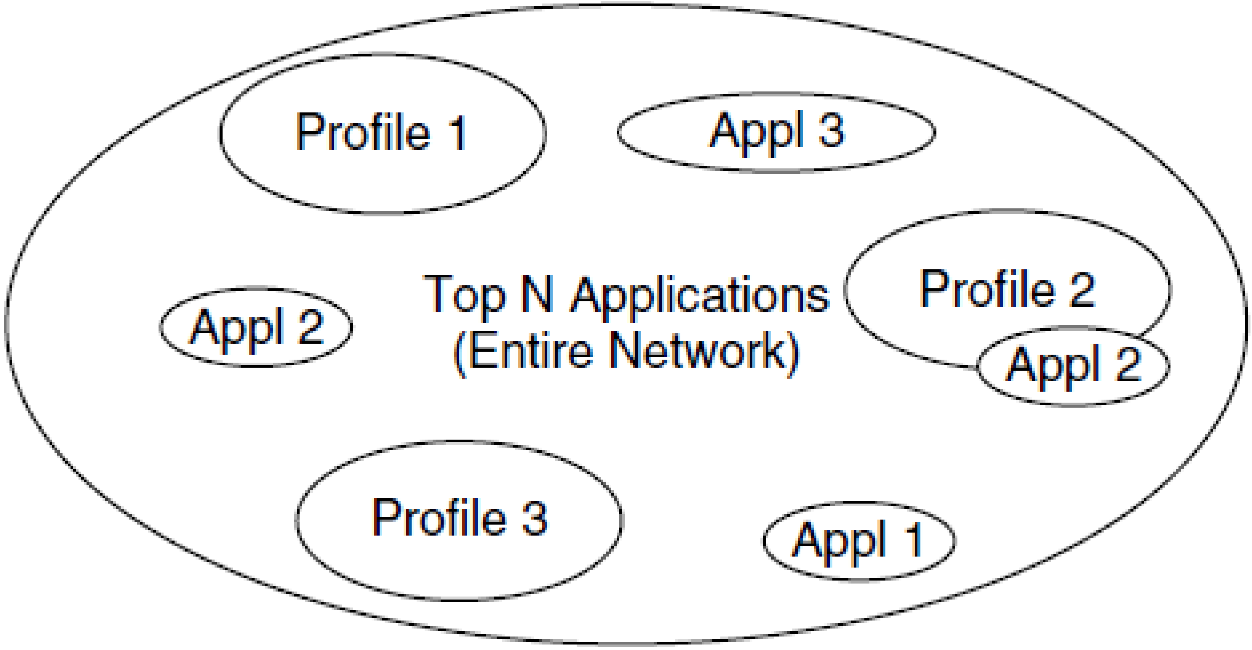
4.3 Flows Identification

4.3.3 Choosing the Top N Application

- Choosing the top N applications is a combination of the first two approaches.
- This approach reduces the set of possible applications to a number that can be analyzed.
- The intent of this approach is to determine which applications represent the most important requirements for that network.

4.3 Flows Identification

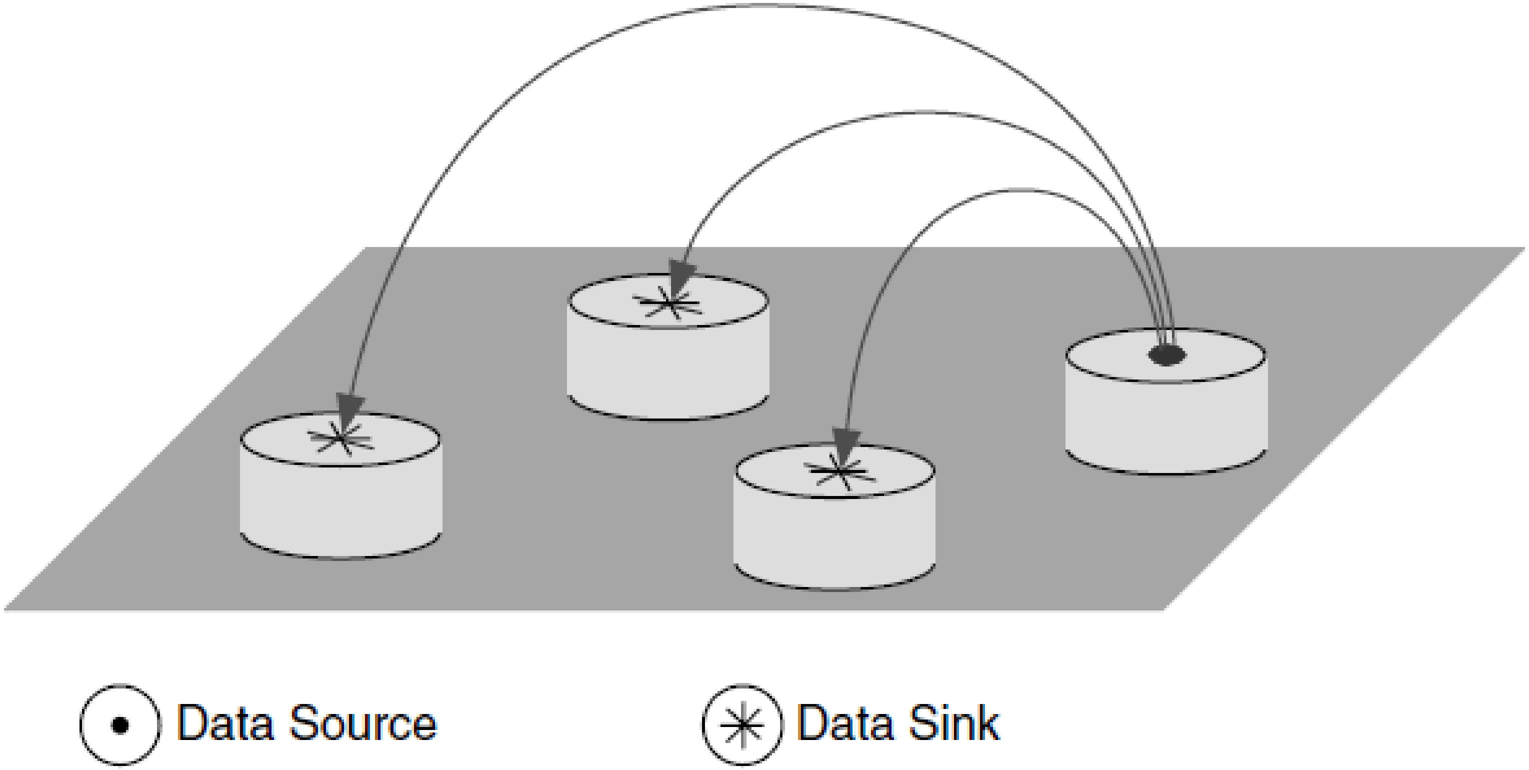
– It may also use different approaches for different parts of the network.



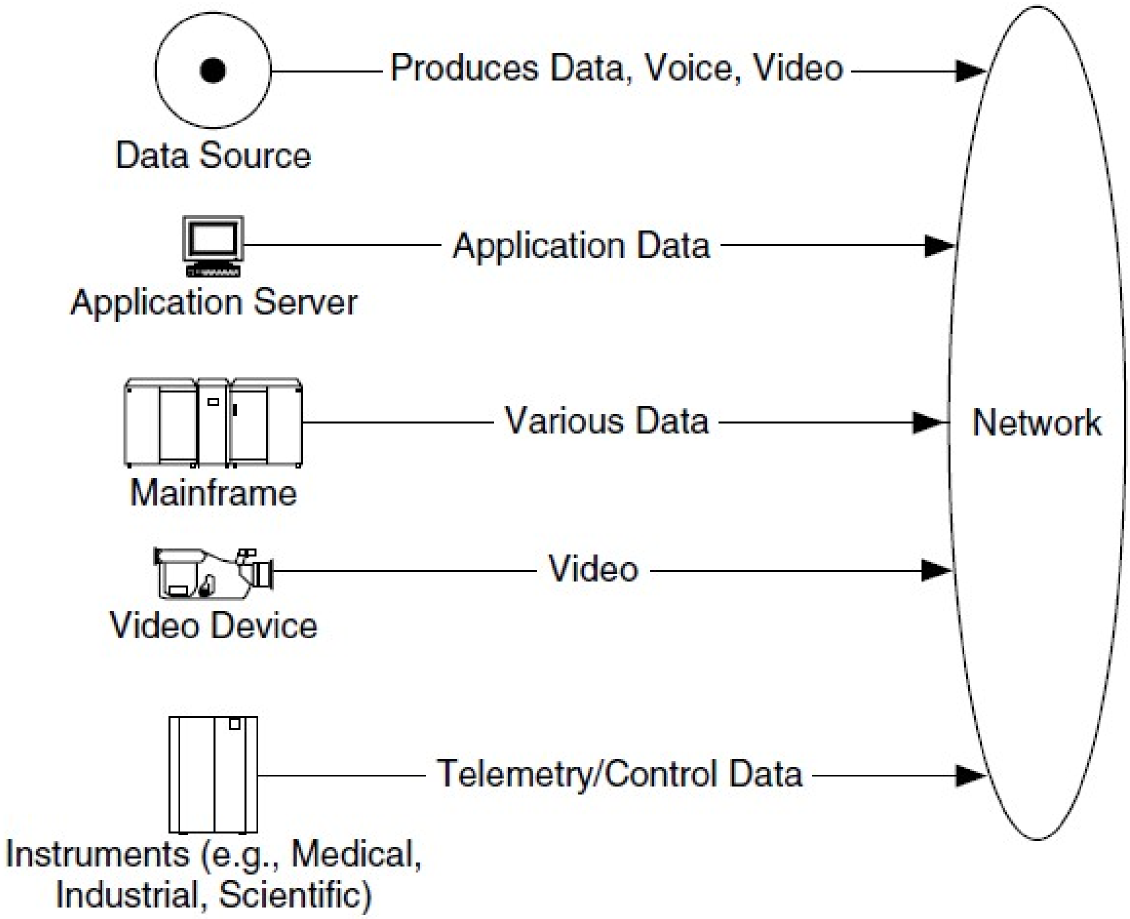
4.4 Data Sources & Sinks

- Data sources and sinks can help provide directionality to flows.
- A *data source* generates a traffic flow, and a *data sink* terminates a traffic flow.
- Data sources are represented as a circle with a dot in the center, and a data sink is represented as a circle with a cross (i.e., star or asterisk) in the center.

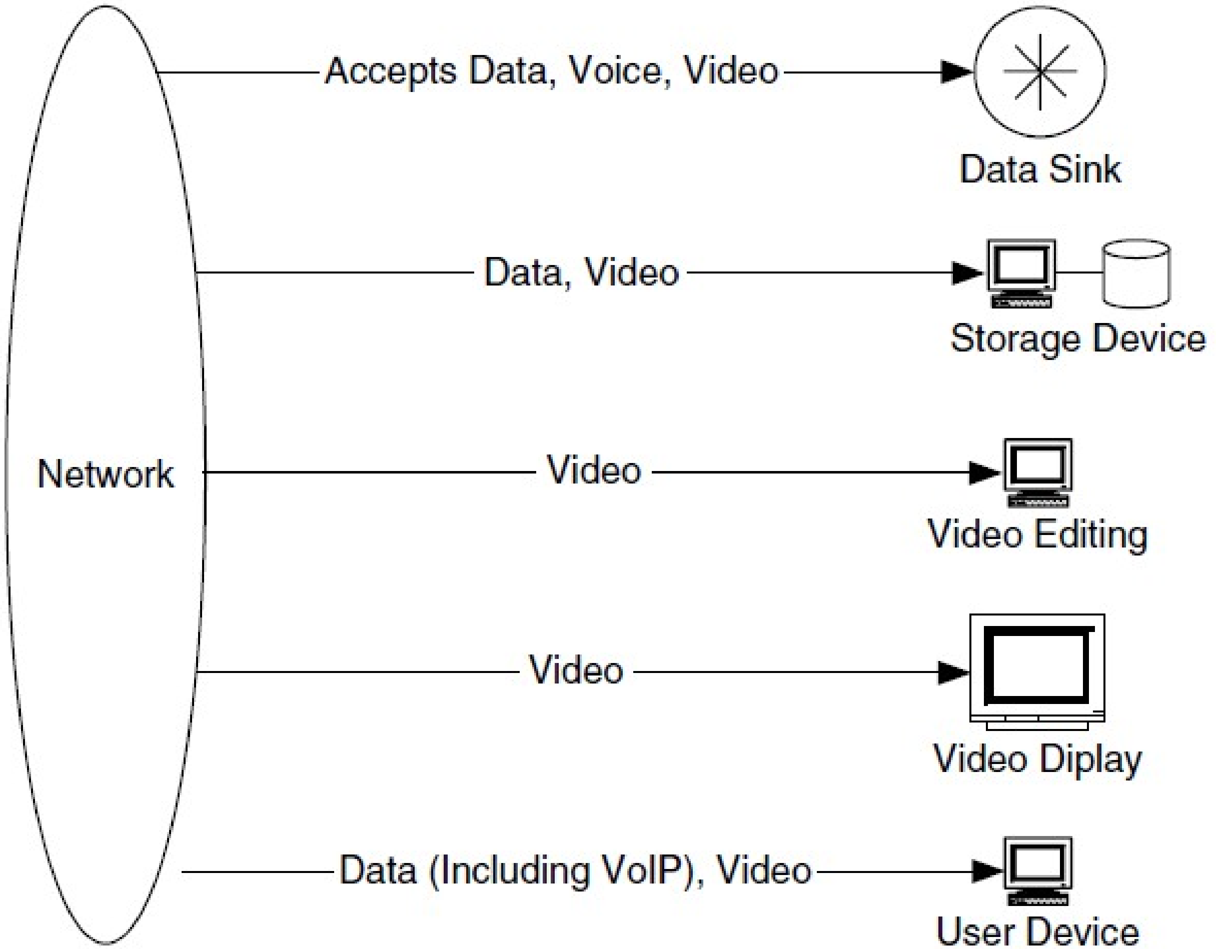
4.4 Data Sources & Sinks



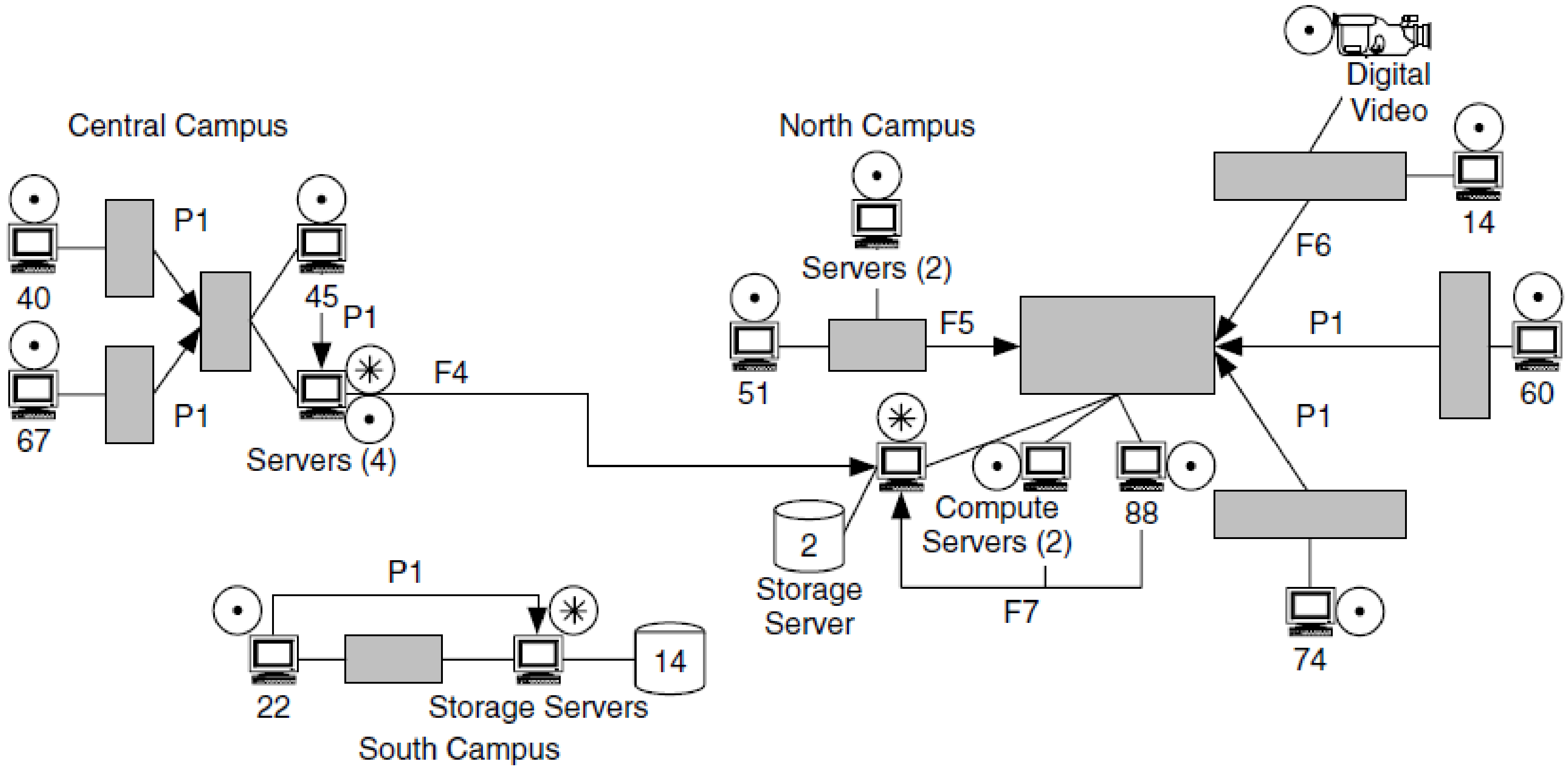
4.4 Data Sources & Sinks



4.4 Data Sources & Sinks



4.4 Data Sources & Sinks



4.5 Flow Models

- *Flow models* are groups of flows that exhibit specific, consistent behavior characteristics.

- The flows within a flow model apply to a single application.

4.5 Flow Models

- Flow models help describe the degrees of hierarchy and diversity of flows for applications.

- Flow models can also be useful to help quickly identify and categorize flows in an environment

4.5 Flow Models

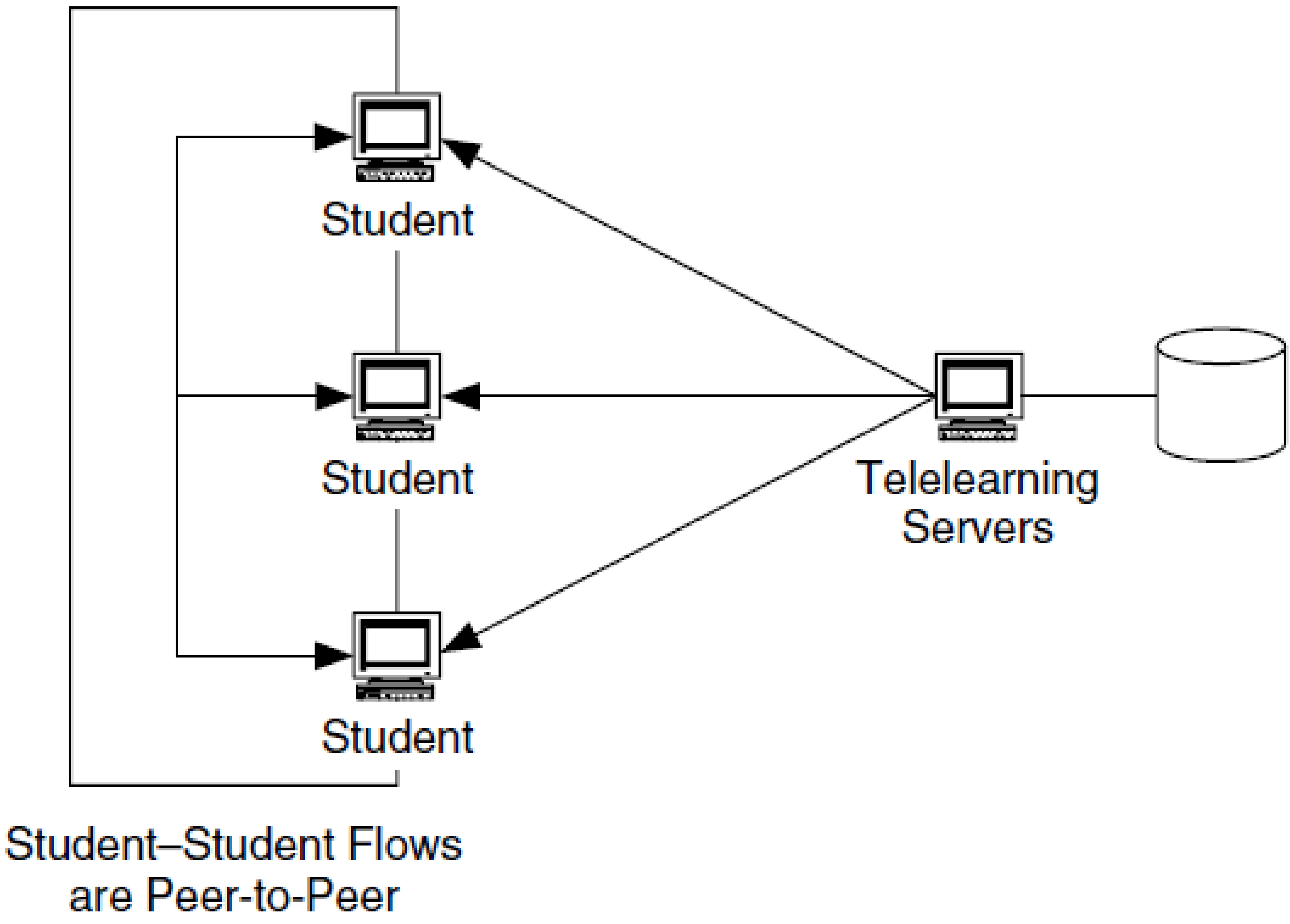
- Flow models could be:
  - Peer-to-peer
  - Client–server
  - Hierarchical client–server
  - Distributed computing

4.5 Flow Models

4.5.1 Peer-to-Peer

- Our first flow model, *peer-to-peer*, is one where the users and applications are fairly consistent in their flow behaviors throughout the network.
- The flows in a peer-to-peer flow model are equivalent,
  - either all of the flows or none of the flows is critical
  - can be described by a single specification (e.g., profile)

4.5 Flow Models



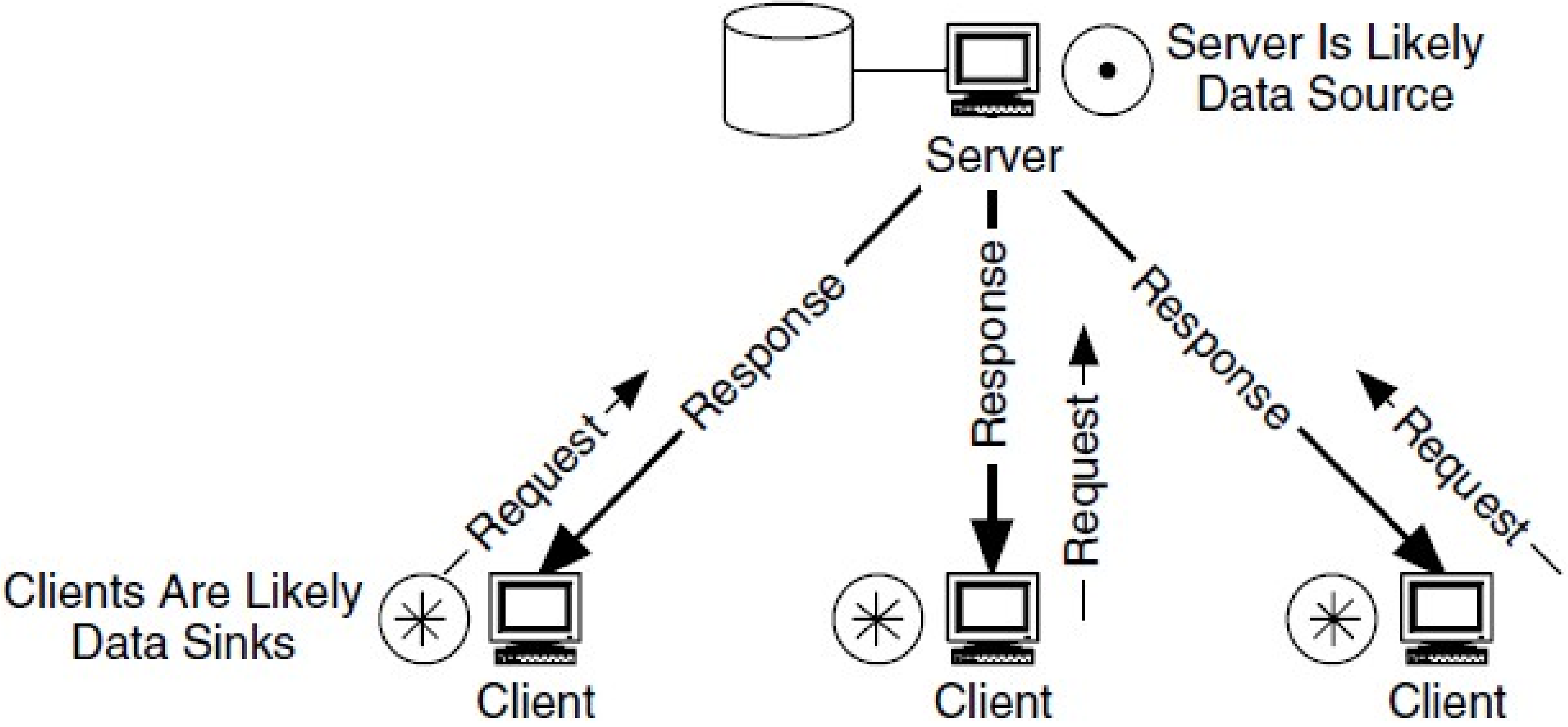
4.5 Flow Models

4.5.2 Client – Server

– This model has both directionality and hierarchy.

– This flow model is *client–server* in that the flows are asymmetric and hierarchically focused toward the client.

4.5 Flow Models

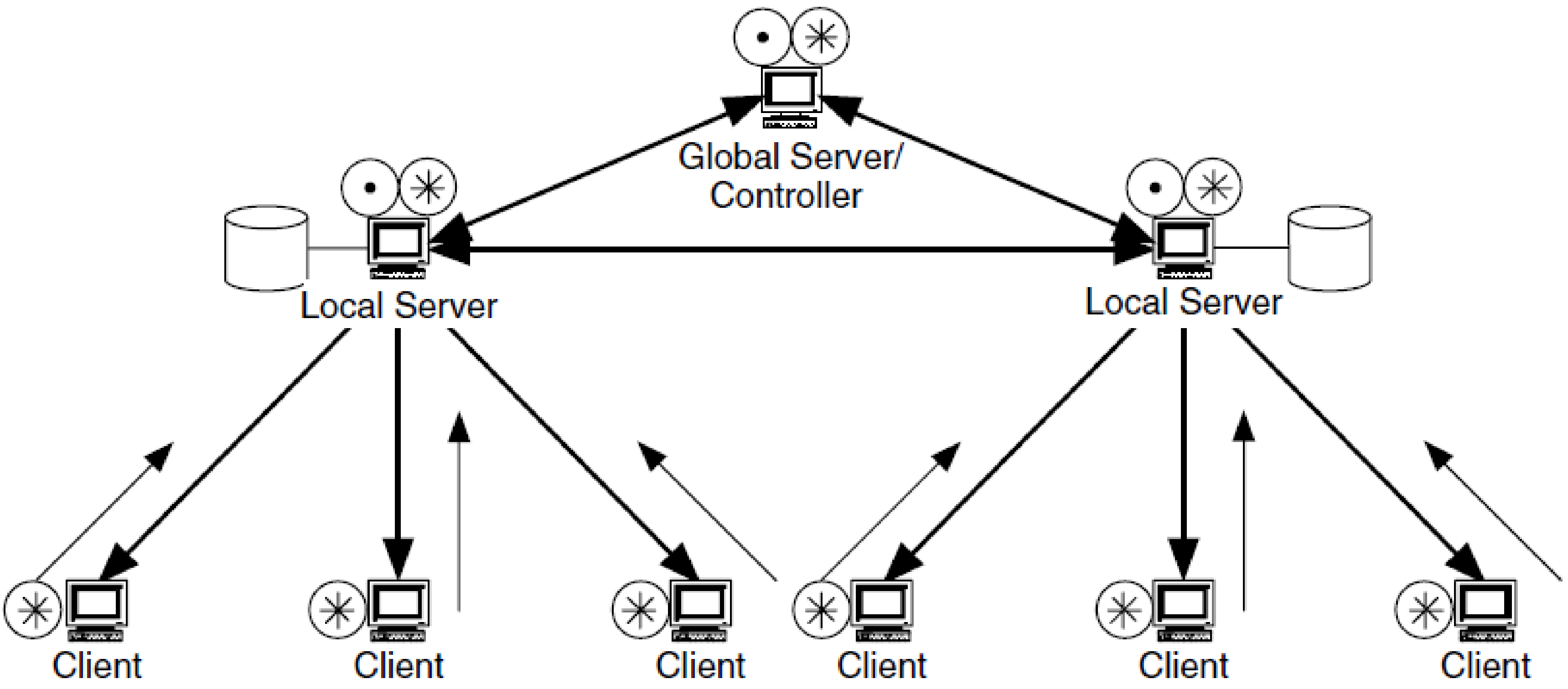


4.5 Flow Models

4.5.3 Hierarchical Client – Server

- A *hierarchical client–server flow model* has the characteristics of a client–server flow model but also has multiple layers, or tiers, between servers.
- It is indicated when multiple applications work together and share information to accomplish a task, or when multiple client–server applications are managed by a higher-level application.

4.5 Flow Models

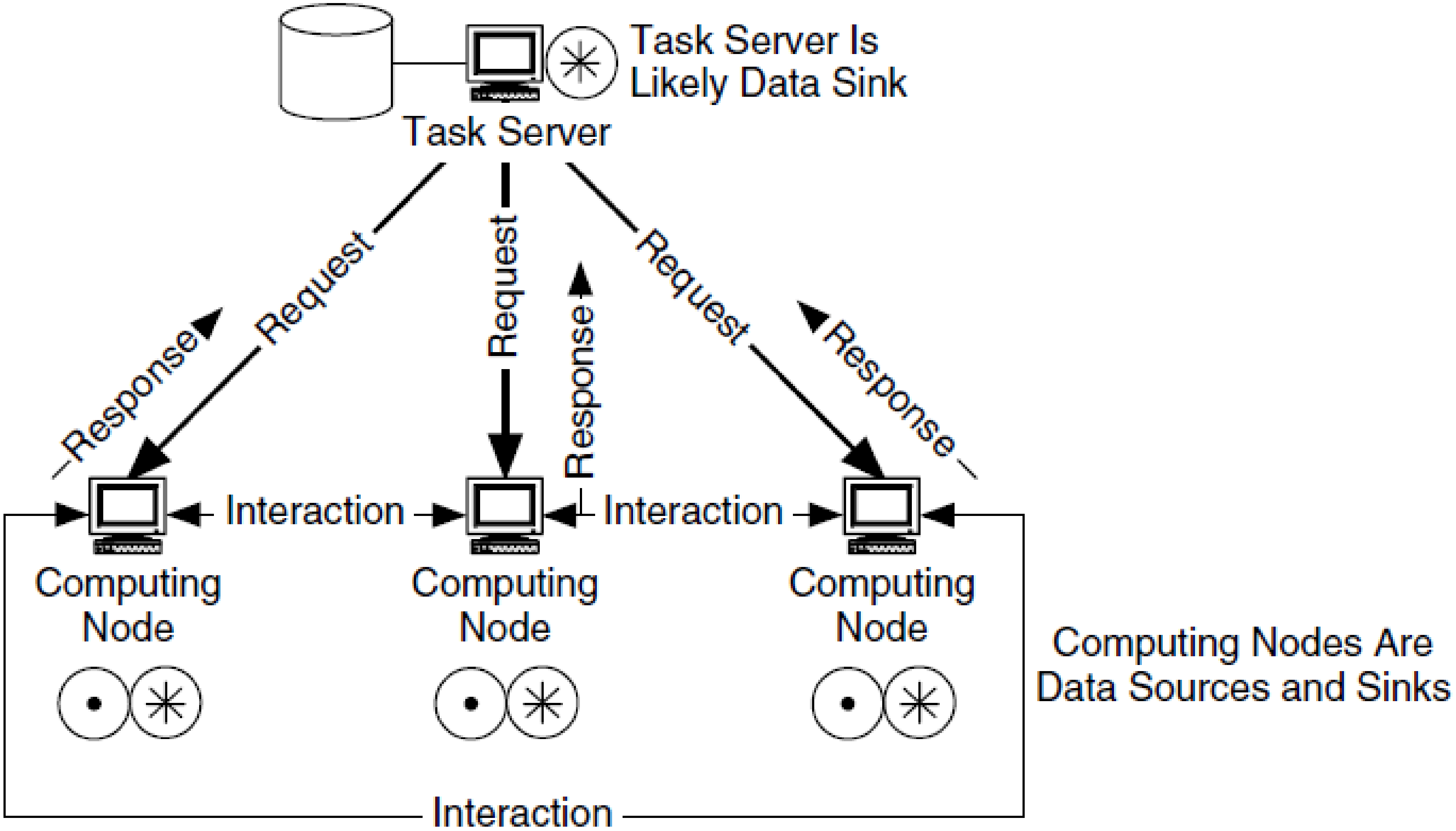


4.5 Flow Models

4.5.4 Distributed – Computing

- A distributed-computing flow model can have the inverse of the characteristics of the client–server flow model, or a hybrid of peer-to-peer and client–server flow models.
- The important characteristics of this model are that the flows can be client–server but are reversed in direction, and that the computing devices may have strict performance requirements.

4.5 Flow Models



4.6 Flow Prioritization

- Flow prioritization means ranking flows based on their importance, which can be described in various ways, depending on the environment.
- Some common prioritizations include:
  - Business objectives and the impact of a flow on the customer's business
  - One or more of the performance requirements of the flow (capacity, delay, RMA, and QoS)

– Security requirements for each flow
– The numbers of users, applications, and/or devices that a flow serves
– Political objectives
– etc.

Flow ID	Performance Requirements			Number of Users
	Reliability	Capacity	Delay	
F1	N/A	1.2 Mb/s	10 ms	1200
F2	99.5%	100 Kb/s	N/A	550
F3	99.5%	15 Kb/s	100 ms	100
CF1	99.95%	500 Kb/s	100 ms	1750
CF2	N/A	100 Kb/s	100 ms	2100
CF3	N/A	3 Mb/s	100 ms	50

4.7 Flow Specification

- The results of identifying, defining, and describing flows are combined into a flow specification, or flow-spec.
- Flow specifications can take one of three types: one-part, or unitary; two-part; or multi-part.

4.7 Flow Specification

- A *one-part flow-spec* describes flows that have only best-effort requirements.

- A *two-part flow-spec* describes flows that have predictable requirements and may include flows that have best-effort requirements.

4.7 Flow Specification

- A *multi-part flow-spec* describes flows that have guaranteed requirements and may include flows that have predictable and/or best-effort requirements

4.7 Flow Specification

Flow Specification Type	Types of Flows	Performance Description
One-Part	Best-Effort Individual and Composite	Capacity Only
Two-Part	Best-Effort and Stochastic, Individual and Composite	Reliability, Capacity, and Delay
Multi-part	Best-Effort, Stochastic, and Guaranteed, Individual and Composite	Reliability, Capacity, and Delay

4.7 Flow Specification

4.7.1 Flowspec Algorithm

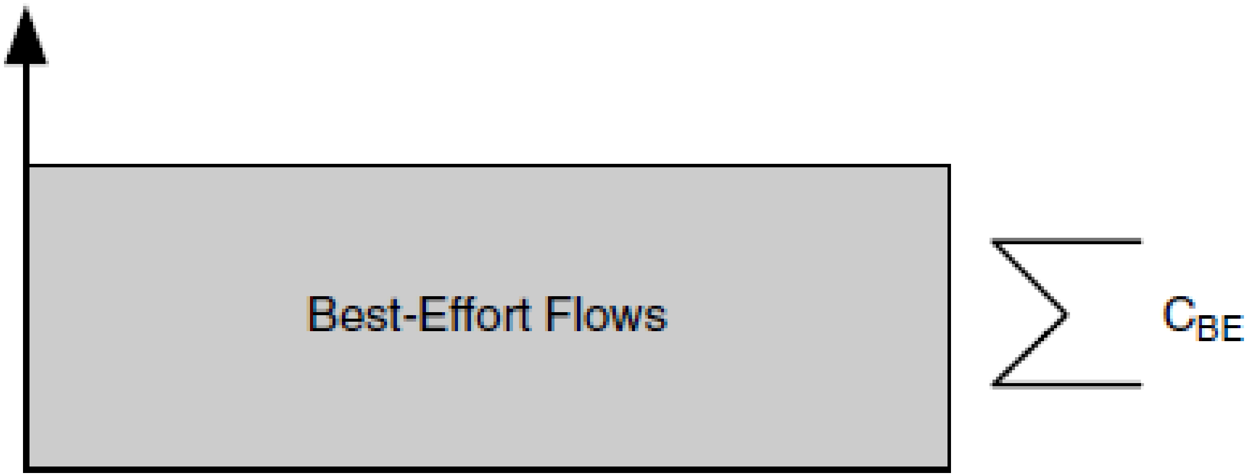
- Flowspecs are used to combine performance requirements of multiple applications for a composite flow or multiple flows in a section of a path.
- The *flowspec algorithm* is a mechanism to combine these performance requirements (capacity, delay, and RMA) for flows in such a way as to describe the optimal composite performance for that flow or group of flows.

4.7 Flow Specification

- The flowspec algorithm applies the following rules:
  - Best-effort flows consist only of capacity requirements; therefore, only capacities are used in best-effort calculations.
  - For flows with predictable requirements, all available performance requirements (capacity, delay, and RMA) are used in the calculations.
  - For flows with guaranteed requirements, each individual requirement are list (as an individual flow), not combining them with other requirements.

4.7 Flow Specification

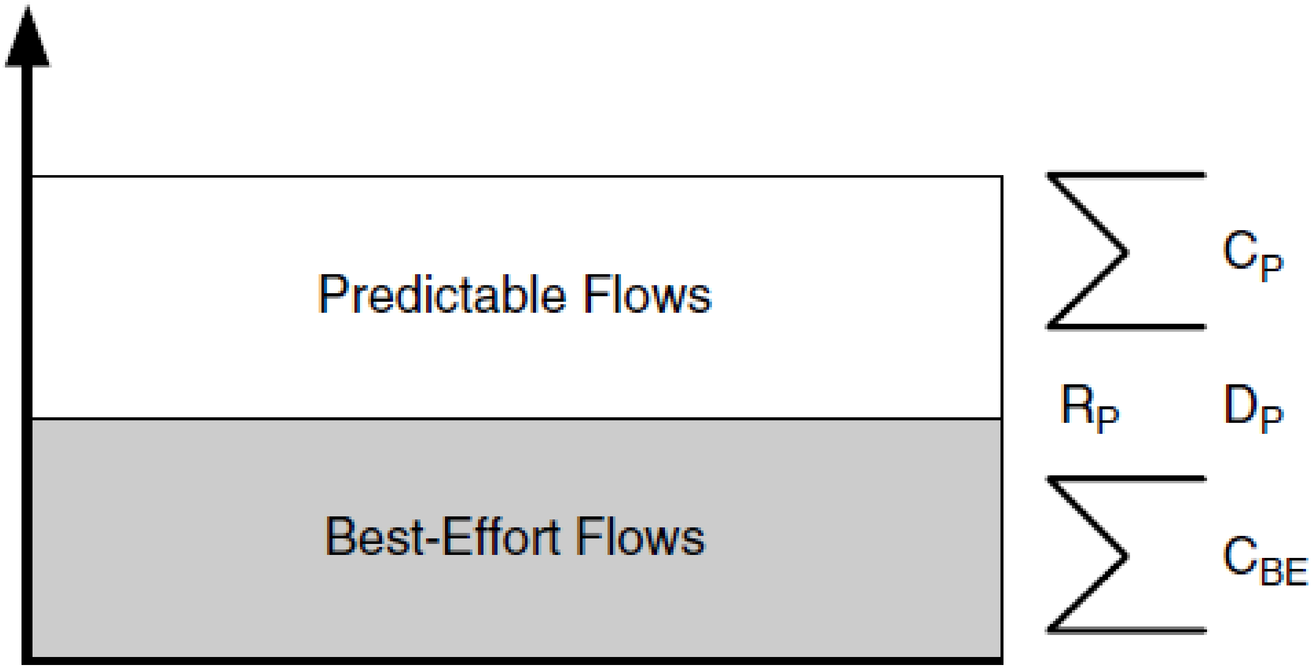
- When a one-part flowspec is developed, capacities are added together, forming a total best-effort capacity C_{BE},



4.7 Flow Specification

- For the predictable requirements, the flowspec has a total capacity for best-effort flows C_{BE} and another capacity for predictable flows C_P
- For delay, the minimum delay is taken as delay requirement D_P, and
- The maximum RMA R_P is taken as RMA requirement

4.7 Flow Specification

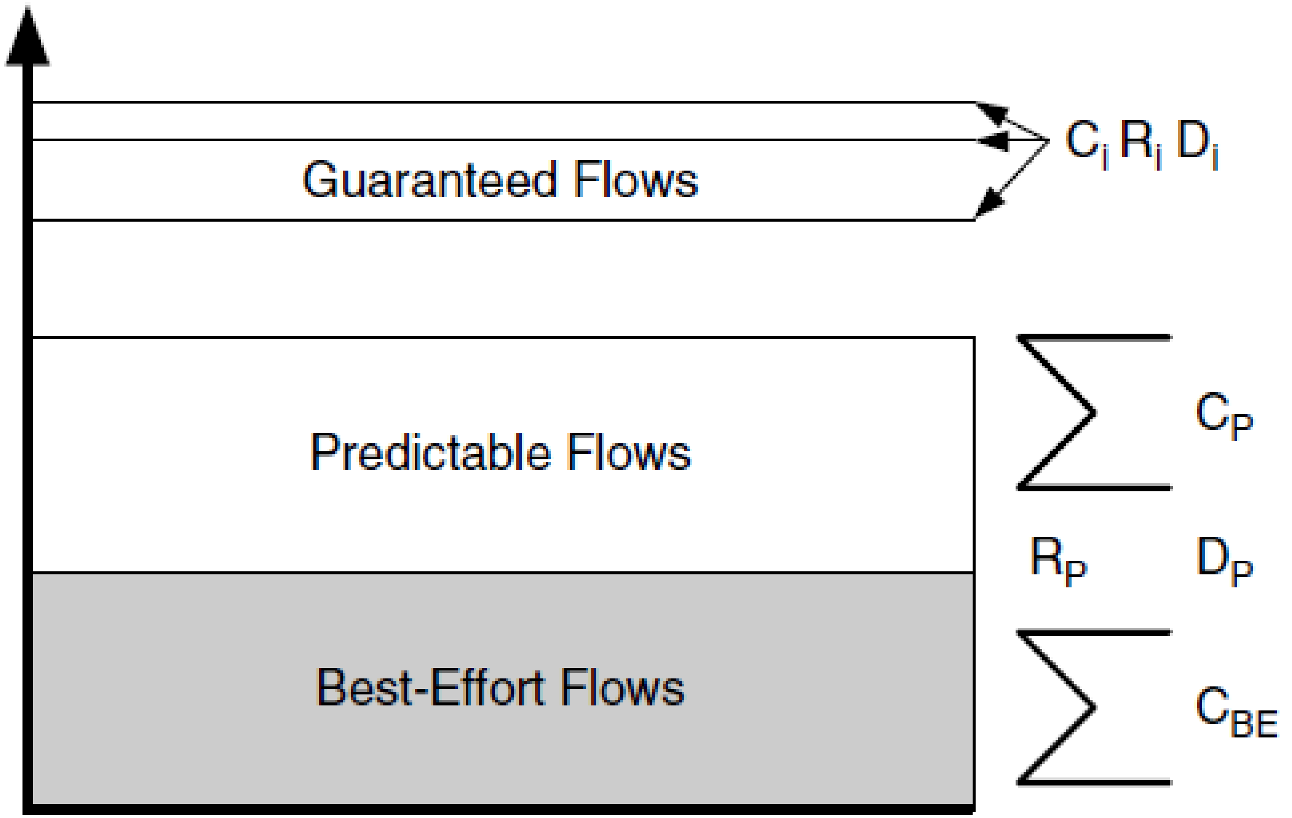


4.7 Flow Specification

– A multi-part flowspec is the most complex of the flowspecs, building on a two-part flowspec to add guaranteed requirements.

– Best-effort capacity, along with predictable capacity, delay, and RMA, and each set i of guaranteed performance requirements is added individually (shown as C_i R_i D_i) to the flowspec.

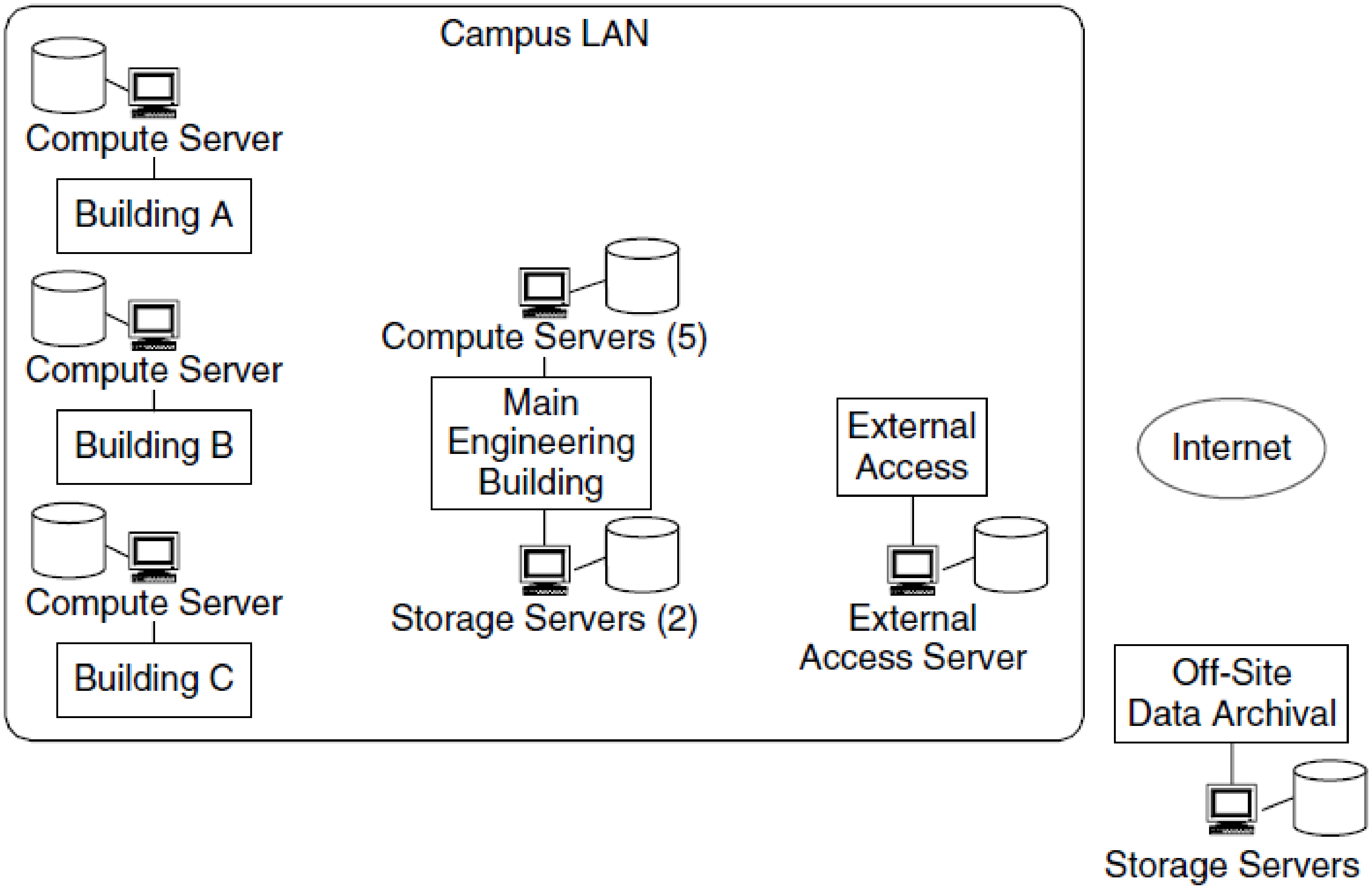
4.7 Flow Specification



4.8 Example Application

- **Case**
  - ***a network to support computing and storage management***
  - ***the computing and storage devices already exist in multiple buildings on a campus***

4.8 Example Application



4.8 Example Application

- **Solution**
  - ***How many flows ?***
  - ***Application Type and Running Mode ?***
  - ***Define Performance Envelope !***
  - ***Building Flow Models !***
  - ***Create Flow Maps***

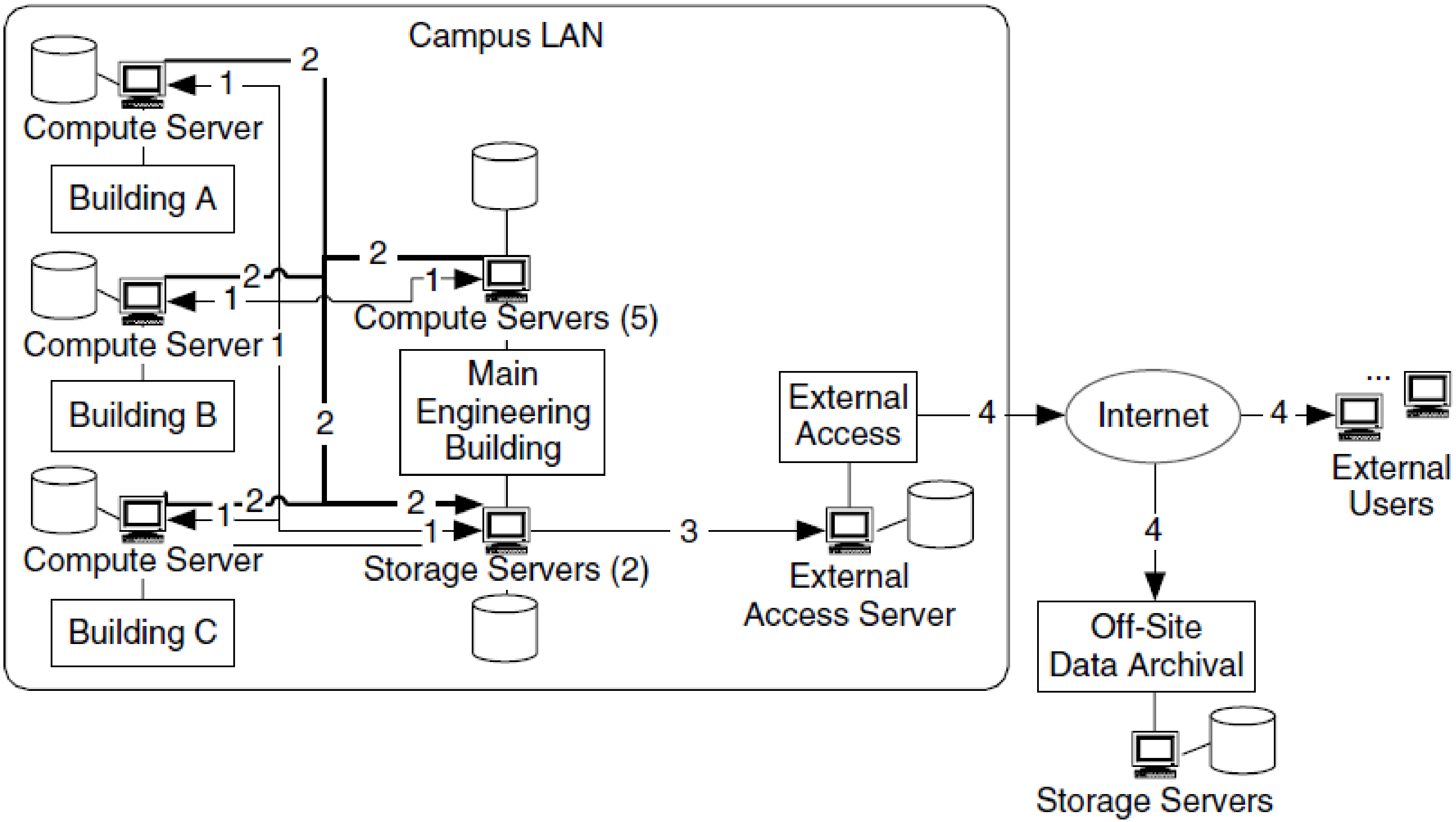
4.8 Example Application

- ***Flows*** :
  - *Type 1: flows between computing devices*.
    - Flows are sometimes (approximately 10% of time) synchronized between pairs of devices.
    - At other times these devices may draw data from the local data store or other computing device.
  - *Type 2: Data migration from computing to storage at Main Engineering*.
    - These flows may be from each compute server to storage server.

4.8 Example Application

- *Type 3: Data migration to external server*.
  - Final data sets, are pushed from storage server to another storage server where external access is managed.
- *Type 4: Data archival and Internet access*.
  - These are flows from the external access server to users on the Internet, as well as flows to off-site archival servers.

4.8 Example Application



4.8 Example Application