MAN Optimization via Constraint-Based Routing in a Heterogeneous Internet of Things

Eric Sturzinger

Group Meeting

30 SEP 2016

UCDAVIS

Outline

- Purpose/Motivation
- Problem Statement
- Latency
- Application Profiles
- Mathematical Description
- Topology/Flow Scenarios
- Simulation
- Ongoing Work

Purpose/Motivation

- Will IoT simply require more physical layer core capacity? Or will traffic nature, i.e. increased heterogeneity require more robust traffic engineering and/or policy based/constraint-based routing?

- Application Heterogeneity:
  - Bandwidth/Latency
  - Processing
  - Storage
  - Internal vs External to MAN

- With increased application/traffic heterogeneity in IoT, effective traffic engineering & function assignment will have a much more significant impact on network costs and performance.

Problem Statement

- As IoT related traffic becomes increasingly heterogeneous and demands greater proportions of overall Internet traffic, it must be segmented according to its unique performance and functional requirements, in this case: latency, throughput, processing, and storage requirements.
- If IoT network deployment and provisioning is not planned and executed in a methodical manner, MAN costs can increase rapidly.

- Given core network SLAs, MAN topology, and performance requirements by app profile: minimize operational costs of MAN via 3 components:
  - Path from generating node to MAN/WAN interface
  - Processing node, if applicable
  - Storage node, if applicable

- Goal is to model mixtures of anticipated application/traffic profiles routed through a MAN with maximum end-to-end latency via constraint-based routing to demonstrate what factors (applications parameters, topological) have a more significant impact on total costs.

Latency

- "This means that we avoid sending all data from sensors and devices back to the cloud, but instead build data and applications on the edge of the network that can handle most of the data gathering and processing. The benefit is better performance and efficiency. IoT applications need to react almost instantly to the data generated by a sensor or device…"

- "Managing and coordinating real-time performance in the IoT will pose a host of new challenges. First and foremost is the problem of scale: this will be a lot more data, coming from lots of different devices. "

- NTT North America – "The average monthly Latency on the NTT Communications Backbone will be 50 milliseconds or less for the North American Network."

- AT&T – "Aggregate monthly average, roundtrip POP-to-POP latency on the IP/DSL Backbone Network shall be 40.0 ms or less between MegaPOP locations on the AT&T IP/DSL Backbone."

http://www.infoworld.com/article/3069591/internet-of-things/we-need-fog-computing-not-cloud-computing-for-iot.html
https://www.wirelessweek.com/article/2015/04/why-latency-management-will-decide-future-iot
http://www.us.ntt.net/support/sla/network.cfm
http://www.att.com/gen/general?pid=6622

UCDAVIS
UNIVERSITY OF CALIFORNIA

Application Profile

- Each application profile contains several parameters, the combination of which make it unique
- α: Computational requirements per unit of traffic
- β: Ratio of post processed to preprocessed data at processing node
- P = 1 if App profile requires processing, S = 1 if App profile requires storage, T = minimum storage tier required by App

App Profile	α	β	P	S	T	Latency
VR	3.0	0.7	1	0	0	80 ms
Indust Data	0.2	0.9	1	1	1	200 ms
Env Data	0	1.0	0	1	3	300 ms
P-P interactive	0.3	0.9	1	0	0	90 ms
Encrypted Data	1.0	1.2	1	0	0	150 ms

Mathematical Formulation

Inputs:

$v_{a,f}^{s,d}$: Offered traffic of application profile **a**, between node pair **(s,d)**, destined for core node **f**

$P_a = 1$ if App profile **a** requires processing of any type

$S_a = 1$ if App profile **a** requires storage of any type

T_a : Set of all nodes capable of storing traffic of App profile **a**

α_a : Computational power required per unit of traffic of App profile **a**

β_a : Ratio of processed traffic to raw traffic of application profile **a**

L_a : Maximum end to end latency for traffic of application profile **a**

$B_{a,f}$: Residual latency budget of traffic destined for core node **f** of application profile **a**

$D_k^{s,d}$: Total latency of k^{th} admissible path of node pair **s,d**

Mathematical Formulation (cont.)

Inputs:

ν_n: Cost of storage per unit of traffic at storage node **n**.

γ_m: Cost per unit computation power of traffic processed at node **m**

$p_k^{s,d}$: $\mathbf{k}^{th}$ admissible path between node pair **s,d**

$q_{k,a,f}^{s,d}$: $\mathbf{k}^{th}$ admissible path between node pair **s,d** which satisfies maximum latency requirements of App profile **a** destined for core node **f**

$d_{q,i,j}$: Queueing delay at node **i** on link **i j**

$d_{n,i}$: Processing delay at node **i** (*processing nodes*)

$d_{t,i,j}$: Transmission delay at node **i** on link **i j**

$d_{p,i,j}$: Propagation delay on link **i j**

Mathematical Formulation (cont.)

Variables:

$x_{a,f}^{s,m} = 1$ if traffic of App profile **a**, destined for core node **f**, generated at source node **s**, is processed at node **m**

$y_a^{s,n} = 1$ if traffic of App profile **a**, gen at node **s**, is stored at node **n**

$r_{a,k,f}^{s,d} = 1$ if traffic of application profile **a** is routed over the $\mathbf{k}^{th}$ admissible path between node pair **(s,d)**, destined for core node **f**

Auxiliary Variables:

$z_a^{s,m,n} = x_{a,0}^{s,m} y_a^{s,n} = 1$ if traffic of App profile **a**, generated at node **s**, is processed at node **m** and stored at node **n**

UCDAVIS
UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Objective Function:

$$min(Cost_p + Cost_s + Cost_u + Cost_d)$$

$$Cost_p = \sum_{m \in N_p} \gamma_m \sum_{a \in A_p \cup A_{sp}} \alpha_a \sum_{s \in G} x_{a,f}^{s,m} v_{a,f}^{s,m}$$ Processing

$$Cost_s = \sum_a \beta_a \sum_{n \in N_a} \nu_n \sum_{s \in G} y_a^{s,n} v_{a,0}^{s,n} + \sum_{a \in A_{sp}} \beta_a \sum_{n \in N_a} \nu_n \sum_{s \in G} \sum_{m \in N_p} z_a^{s,m,n} v_{a,0}^{s,m}$$ Storage

source to storage — proc to storage, source to proc/storage

$$Cost_u = \epsilon_{up} \Big[\sum_{a \in A_p} \beta_a \sum_{m \in N_l} \sum_{s \in G} \sum_{f \in F_c} x_{a,f}^{s,m} v_{a,f}^{s,m} + \sum_{a \in A_{sp}} \beta_a \sum_{m \in N_l} \sum_{s \in G} z_a^{s,m,28} v_{a,0}^{s,m} +$$

local processing to destination — local proc, storage at DC

$$\sum_{a \in A_n} \sum_{d \in C} \sum_{s \in G} \sum_f v_{a,f}^{s,d} + \sum_{a \in A_p} \sum_{s \in G} \sum_f x_{a,f}^{s,28} v_{a,f}^{s,28} \Big]$$ Upstream

Tx to destination — Proc at DC, back to metro/dest

UCDAVIS UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Objective Function (cont.):

$$Cost_d = \epsilon_{down}\left[\sum_{a\in A_n}\sum_{s\in C\cup DC}\sum_{d\in G}\sum_{f\in F_c} v_{a,f}^{s,d} + \sum_{a\in A_p}\beta_a\sum_{s\in G}\sum_{d\in G} x_{a,3}^{s,28} v_{a,3}^{s,28}\right]$$ Downstream

From distant core node or DC — Proc at DC, back to metro

Constraints:

$$\sum_{m\in N_p} x_{a,f}^{s,m} = P_a, \forall\ (a, f, s)$$

Processing

$$\sum_{a\in A_p}\alpha_a\sum_{s\in G}\sum_{f} x_{a,f}^{s,m} v_{a,f}^{s,m} \le U_m \quad \forall m \in N_p$$

$$\sum_{n\in T_a} y_a^{s,n} = S_a, \forall\ (a, s)$$

Storage

$$\sum_{a\in A_s}\alpha_a\sum_{s\in G}\sum_{s\in G} y_a^{s,n} v_{a,0}^{s,n} + \sum_{a\in A_{sp}}\alpha_a\beta_a\sum_{s\in G}\sum_{m\in N_p} z_a^{s,m,n} v_{a,0}^{s,m} \le E_n,\ \forall n \in N_s$$

source to storage — proc to storage, source to proc/storage

UCDAVIS
UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Constraints (cont.): Capacity

$$\sum_{a\in A_p}\sum_{s\in G}\sum_{m\in N_p} x_{a,f}^{s,m}\sum_{f}\sum_{k\in R_{i,j}} r_{a,k,f}^{s,m} = C_1$$

source to proc

$$\sum_{a\in A_s}\sum_{s\in G}\sum_{n\in T_a} y_a^{s,n}\sum_{k\in R_{i,j}} r_{a,k,0}^{s,n} = C_2$$

source to storage

$$\sum_{a\in A_p}\beta_a\sum_{s\in G}\sum_{m\in N_p}\sum_{d\in C\cup DC}\sum_{f} x_{a,f}^{s,m}\sum_{k\in R_{i,j}} r_{a,k,f}^{m,d} = C_3$$

proc to dest

$$\sum_{a\in A_n}\sum_{s}\sum_{d}\sum_{f}\sum_{k\in R_{i,j}} r_{a,k,f}^{s,d} = C_4$$

tx to dest

$$\sum_{a\in A_{sp}}\beta_a\sum_{s\in G}\sum_{m\in N_p}\sum_{n\in T_a} z_a^{s,m,n}\sum_{k\in R_{i,j}} r_{a,k,0}^{m,n} = C_5$$

proc to storage

$$\sum_{e} C_e \le C_{i,j}, \forall(i,j)$$

UCDAVIS UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Constraints (cont.): Solenoidality

$$\sum_k r^{s,m}_{a,k,f} = x^{s,m}_{a,f} v^{s,m}_{a,f}, \forall (s \in G, m \in N_p, a \in A_p, f)$$ source to proc

$$\sum_k r^{m,n}_{a,k,0} = z^{s,m,n}_a \beta_a v^{s,m}_{a,0}, \forall (s \in G, m \in N_p, n \in T_a \neq m, a \in A_{sp})$$ proc to storage

$$\sum_k r^{s,n}_{a,k,0} = z^{s,m,n}_a v^{s,n}_{a,0}, \forall (s \in G, n \in T_a, m \in N_p, n = m, a \in A_{sp})$$ source to s/p

$$\sum_k r^{s,n}_{a,k,0} = y^{s,n}_a v^{s,n}_{a,0}, \forall (s \in G, n \in T_a, a \in A_s)$$ source to storage

$$\sum_k r^{s,d}_{a,k,f} = v^{s,d}_{a,f}, \forall (s, d, f, a \in A_n)$$ no proc, no stor

$$\sum_k r^{m,d}_{a,k,f} = x^{s,m}_{a,f} \beta_a v^{s,m}_{a,f}, \forall (s, m \in N_p, d, f, a \in A_p)$$ proc to dest

UCDAVIS
UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Constraints (cont.): Latency

$$\sum_{(i,j)\in p_{k,a,f}^{s,d}} (d_{t,i,j} + d_{p,i,j} + d_{n,i,j} + d_{q,i,j}) \leq B_{a,f} \ \forall (k, a \in An, f, s, d)$$

source to dest

$$x_{a,f}^{s,m} \Big(\sum_{(i,j)\in p_{k,a,f}^{s,m}} D_{k,s,m} + \sum_{(i,j)\in p_{k,a,f}^{m,d}} D_{k,m,d} \Big) \leq B_{a,f}$$
$$\forall (k, a \in A_p, f, s \in G, m \in N_l, d \in C \cup DC)$$

Source to local proc to dest

$$y_a^{s,n} \sum_{(i,j)\in p_{k,a,0}^{s,n}} D_{k,s,n} \leq B_{a,0} \ \forall (k, a \in A_s, s \in G, n \in T_a)$$

source to storage

$$z_a^{s,m,n} \Big(\sum_{(i,j)\in p_{k,a,0}^{s,m}} D_{k,s,m} + \sum_{(i,j)\in p_{k,a,f}^{m,n}} D_{k,m,n} \Big) \leq B_{a,f}$$
$$\forall (k, a \in A_{sp}, s \in G, m \in N_l, n \in T_a, n \neq m)$$

Source to proc to storage

UCDAVIS
UNIVERSITY OF CALIFORNIA

Mathematical Formulation (cont.)

Constraints (cont.): Latency

$$z_a^{s,m,n} \sum_{(i,j)\in p_{k,a,0}^{s,m}} D_{k,s,m} \le B_{a,0} \ \forall (k, a \in A_{sp}, s \in G, m \in N_l, n = m)$$

Source to proc/storage

Mathematical Explanation

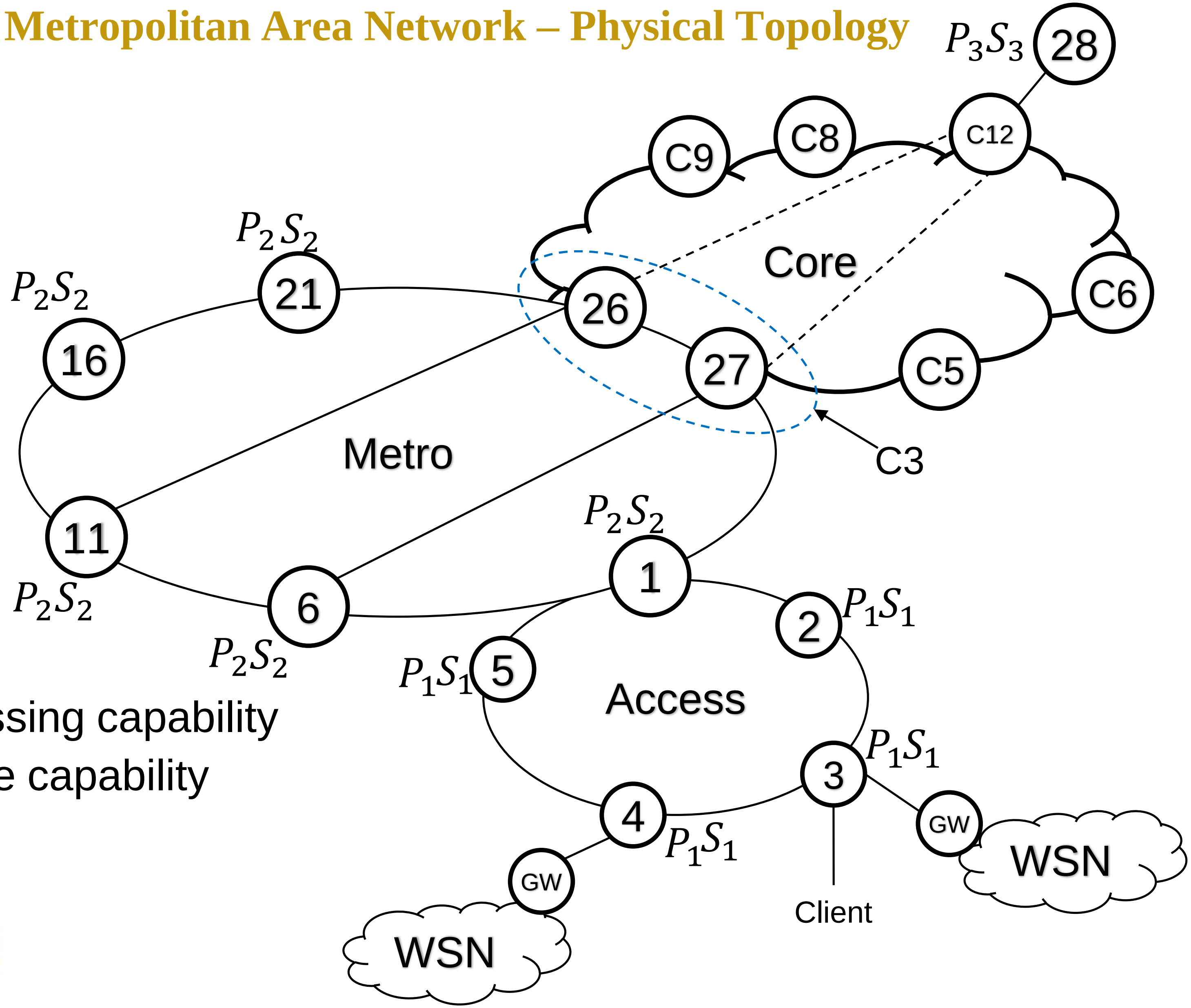
- Inputs:
  - MAN Topology: G(N,L)
    - Node tier dictates costs and capacities for processing and storage – γ, ν, U, E
  - Single cloud storage/DC location
  - Link Capacities
  - Application Profiles:
    - α,β, P, S, T, latency
  - Core Network SLAs

- Objective function: Minimize total operational cost of MAN
  - Processing, Storage, Upstream, Downstream, Capacity

- Outputs: For all node pairs, profiles, and core destinations:
  - Path
  - Processing node (if applicable)
  - Storage node (if applicable)

UCDAVIS
UNIVERSITY OF CALIFORNIA

Metropolitan Area Network – Physical Topology



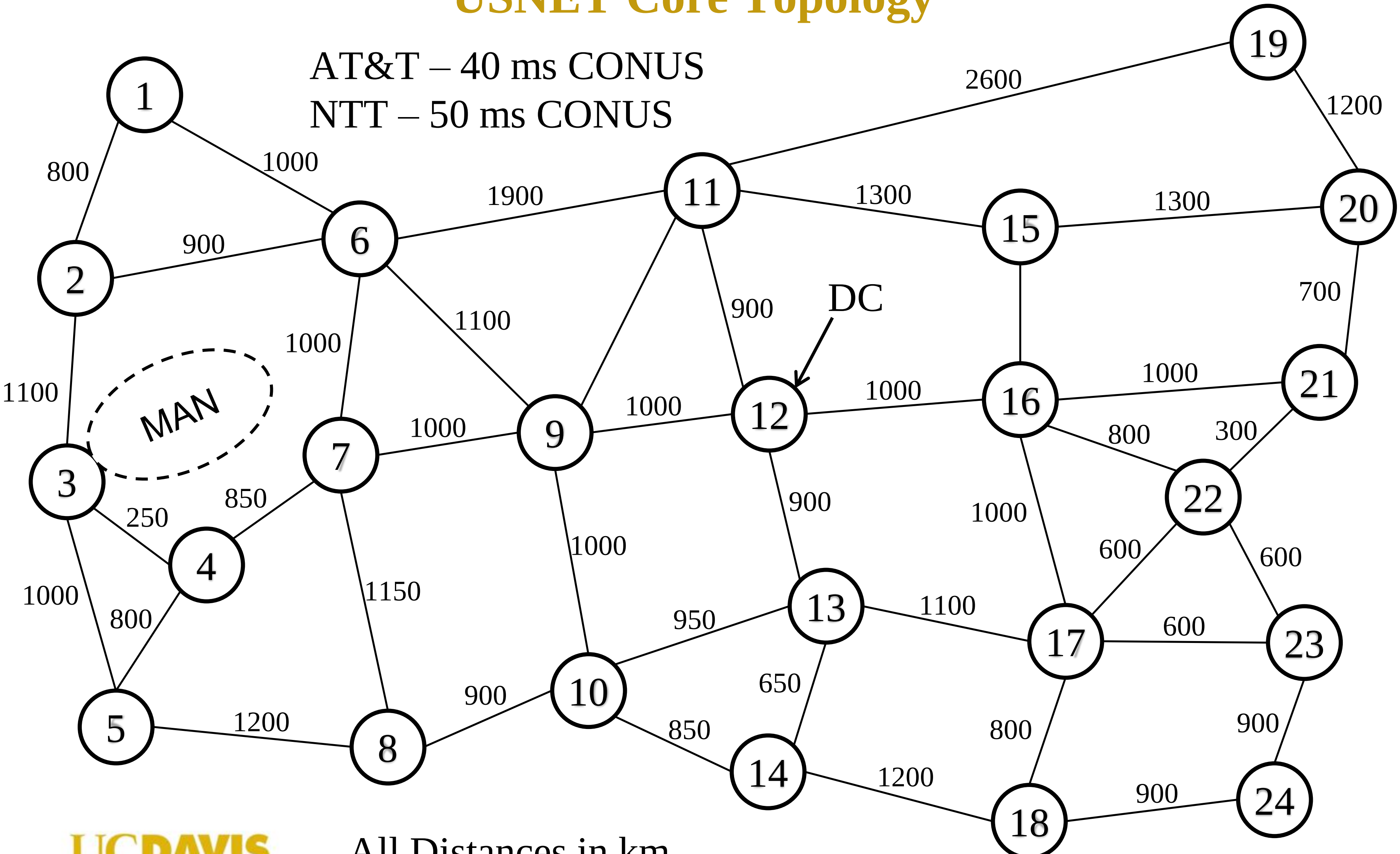
P_t: Tier **t** processing capability

S_t: Tier **t** storage capability

UCDAVIS
UNIVERSITY OF CALIFORNIA

USNET Core Topology

AT&T – 40 ms CONUS
NTT – 50 ms CONUS



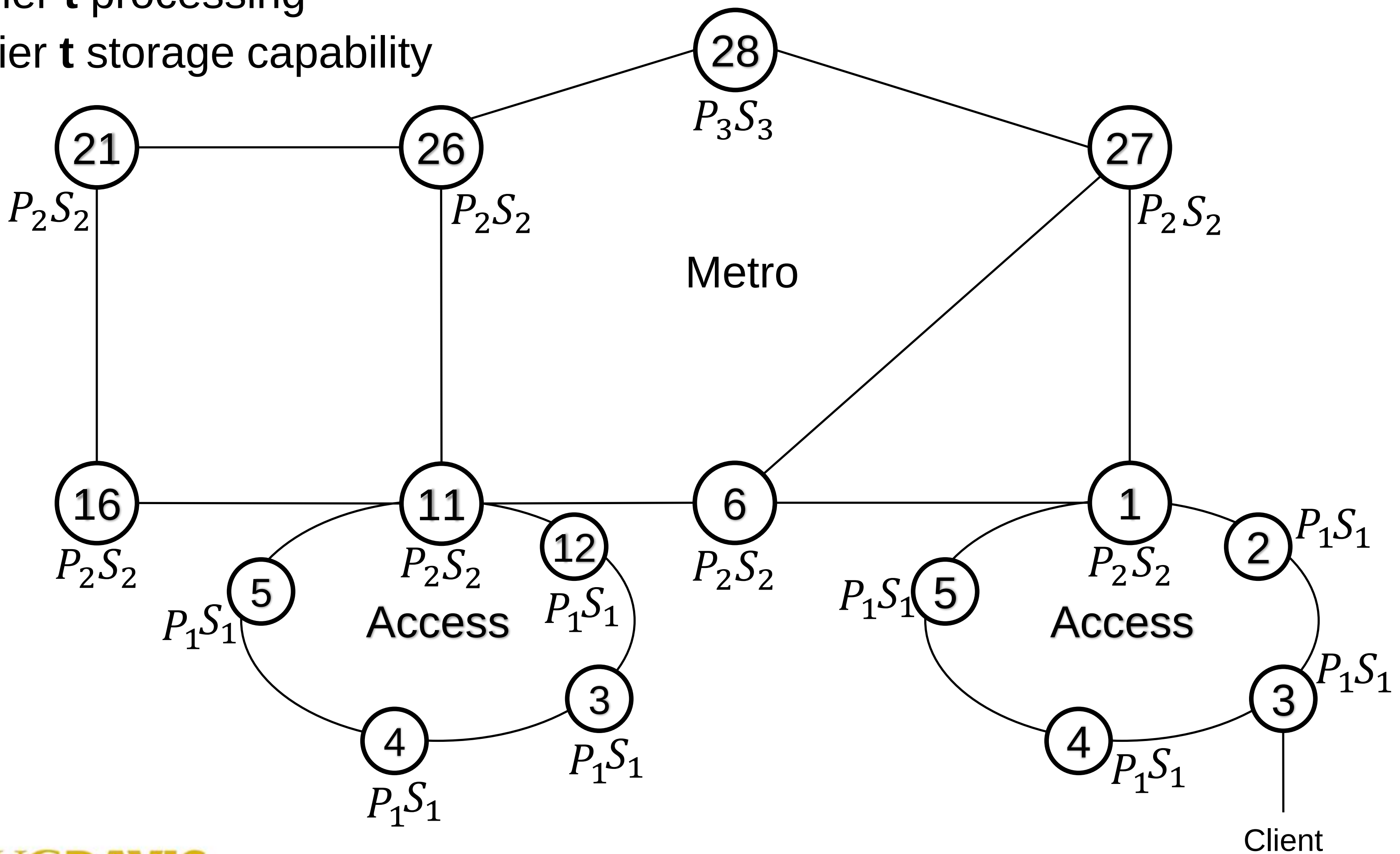
All Distances in km

UC DAVIS
UNIVERSITY OF CALIFORNIA

Metropolitan Area Network – Logical Topology

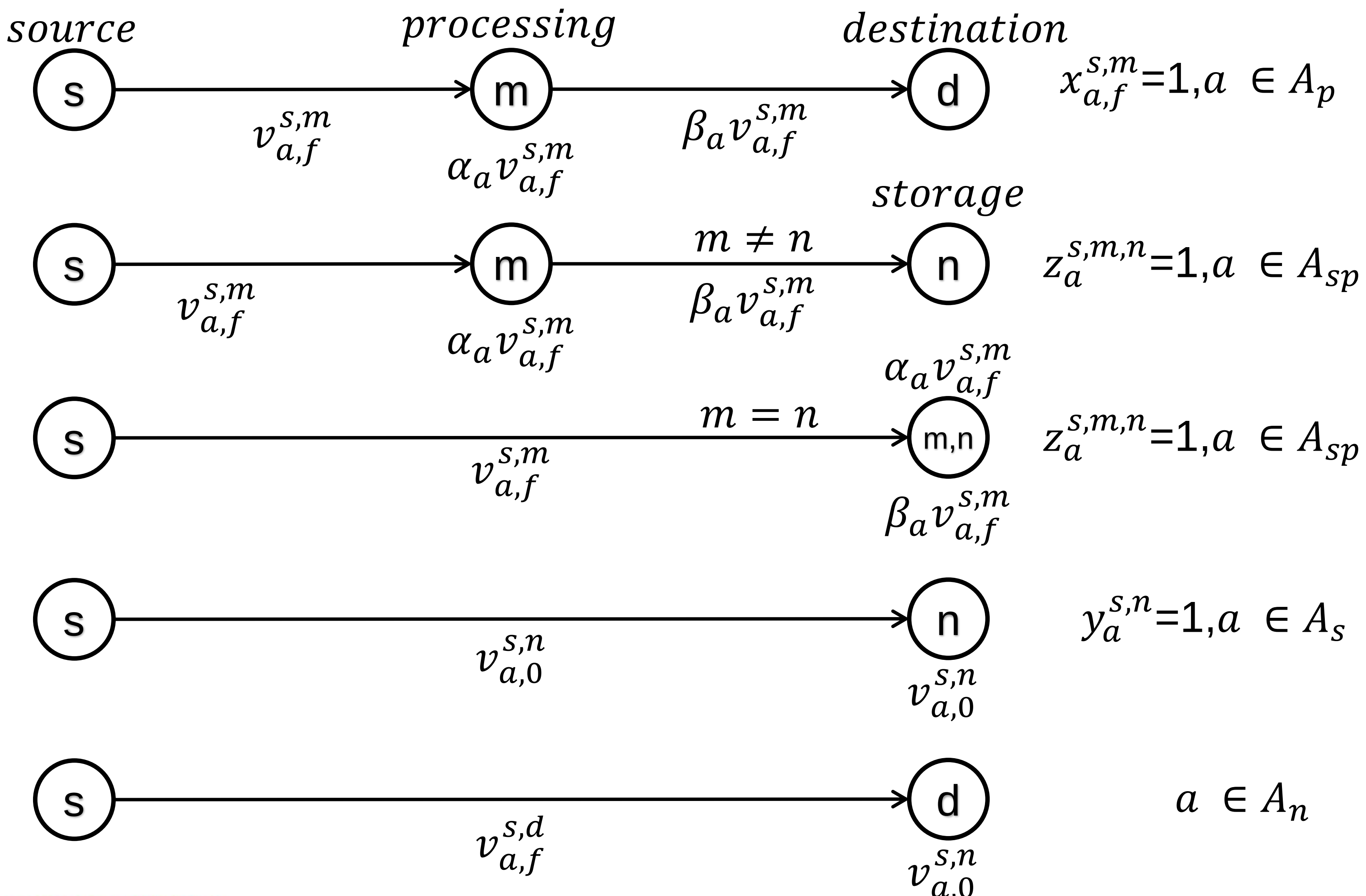
P_t: Tier **t** processing

S_t: Tier **t** storage capability



UC DAVIS
UNIVERSITY OF CALIFORNIA

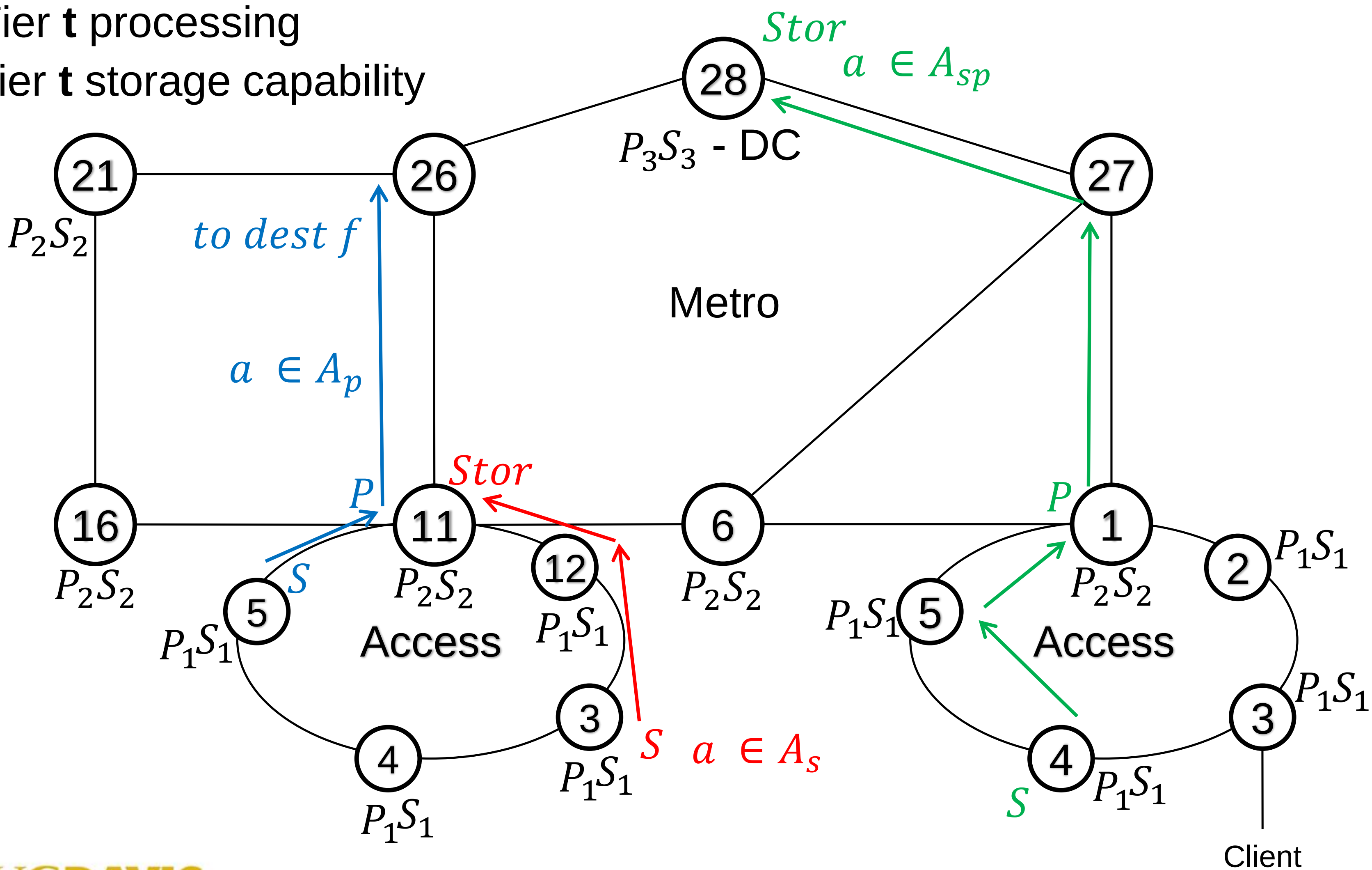
Functional Scenarios



MAN - Traffic Scenarios

P_t: Tier **t** processing

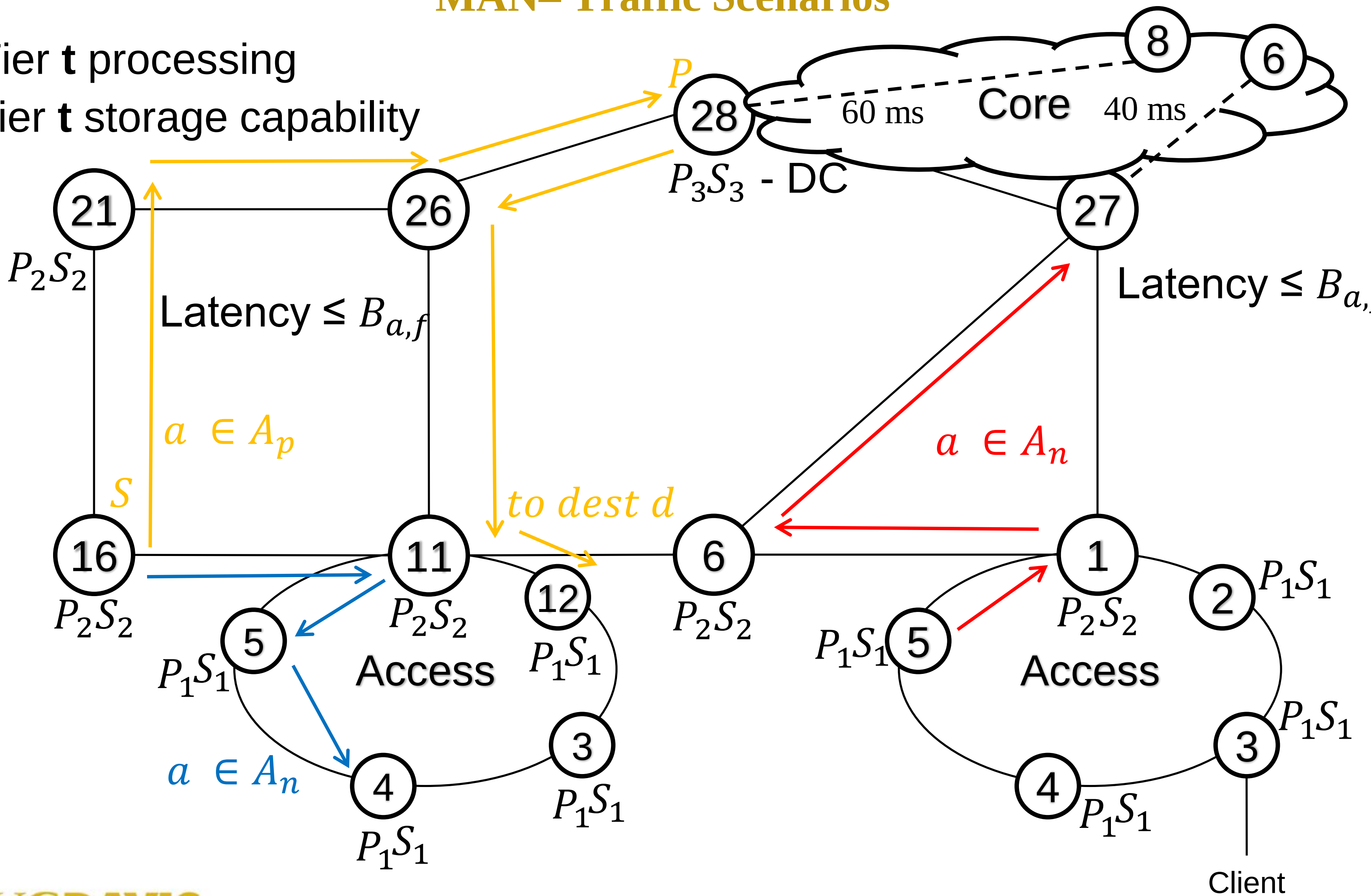
S_t: Tier **t** storage capability



MAN– Traffic Scenarios

P_t: Tier **t** processing

S_t: Tier **t** storage capability



Simulation Strategy

- Generate K shortest paths $\longrightarrow$ Subset of paths which satisfy latency reqs
- Initial fixed link capacities
- Determine what effect the following have on costs:
  - Processing capacities/costs at each tier
  - Storage capacities/costs at each tier
  - Link capacities/increase utilization with offered traffic

- Processing delay:
  - Calculated from initial solution – increases linearly w/ total amount of processed traffic at each node
  - Feed into total path delay of next iteration, compare solutions, continue
  - Similar to flow deviation algorithm

UCDAVIS
UNIVERSITY OF CALIFORNIA

Ongoing Work

- Python modeling: pre-calculation of suitable paths with regard to latency budgets; can simplify problem using single core latency for all destinations
  - Testing individual constraints for syntactic and mathematical errors

- Model queueing delay: Poisson and Power Law Queueing theory equations as feedback into path delay calculations
  - Calculate for each link and include with total delay (dependent on variable) – use as feedback into succeeding iteration
- Model processing delay:
  - Include processing delay as part of total delay – increases linearly with amount of traffic processed by node (dependent on variable)

- Extension: Storage locations can be viewed as potential multicast sources that are accessible from clients distributed across the world
  - Can extend problem to better decide storage locations or possibly move storage analysis from metro to distributed across core

UCDAVIS
UNIVERSITY OF CALIFORNIA