(ICC 2021)

Optimal Cloud Network Control with Strict Latency Constraints

Yang Cai, Jaime Llorca, Antonia M. Tulino, Andreas F. Molisch

Email: yangcai@usc.edu

USC Viterbi School of Engineering

University of Southern California

Background

- Increasing demand for computational resource
  - Real-time computer vision, multi-user conferencing, and augmented/virtual reality
- Limited local computational resource at UE
  - Tendency: light weight, portable devices
  - Restricted processing capability, battery
- Solution: requesting computing service from the cloud
  - Better delay and cost performance

Background

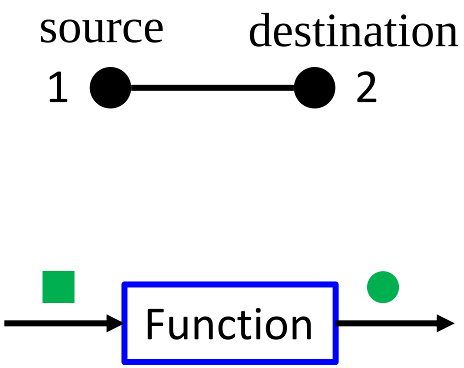
- Distributed cloud network
  - Make it easier for the UEs to access the computational resource
    - Traditional processing network: separation of network & processing center
    - Distributed cloud network: deploy the computational resource in a more widespread manner
- NFV & SDN-enabled Next-Gen Cloud
  - Make it more flexible for the cloud to process the data-stream
    - Computing task → service function chain
    - Each individual function can be implemented separately (at different network locations)

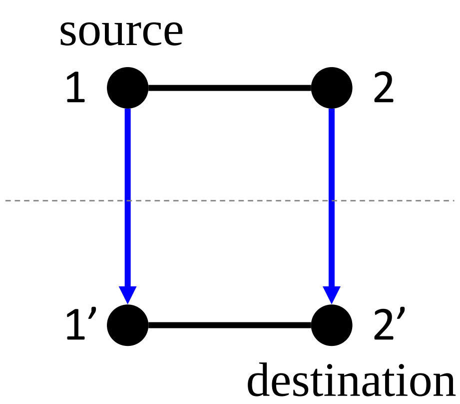
Background

- The goal is to design a dynamic cloud control algorithm that achieves:
  - Better delay performance
    - Autonomous transportation, machine control in Industry 4.0
    - From average delay to per-packet delay
  - Better cost performance
    - Especially in heterogeneous network

System Model

- Cloud layered graph
  - The original problem can be transformed to packet routing problem





System Model

- Request model
  - Lifetime
    - The deadline by which the packet becomes outdated
    - The packet is called effective otherwise
    - I.I.D. arrival processes of packets with various initial lifetime at any node
  - Timely throughput
    - The rate of effective packet delivery
  - Reliability
    - The ratio of effective packets to all arrival packet

System Model

- Queuing system
  - Queues $\boldsymbol{Q}(t) = [Q_i^{(l)}(t)]$
    - The queue of lifetime l at node i on time slot t
  - Flow variables $\boldsymbol{x}(t) = [x_{ij}^{(l)}(t)]$
    - The actual amount of lifetime l packets sent from node i to j
  - Queuing dynamics

exogenous packets

$$Q_i^{(l)}(t+1) = Q_i^{(l+1)}(t) - x_{i\to}^{(l+1)}(t) + x_{\to i}^{(l+1)}(t) + a_i^{(l)}(t)$$

$$Q_i^{(0)}(t) = 0 \quad (\forall\, i \in \mathcal{V})$$

$$Q_d^{(l)}(t) = 0 \quad (\forall\, l \in \mathcal{L})$$

System Model

- Policy space
  - Decision variable: the flow variables $\boldsymbol{x}(t)$
  - Constraints
    - Non-negativity $\boldsymbol{x}(t) \succeq 0$
    - Link capacity constraint $\overline{\{\mathbb{E}\{x_{ij}(t)\}\}} \leq C_{ij}$
    - Availability constraint $x_{i\rightarrow}^{(l)}(t) \leq Q_i^{(l)}(t)$
    - Reliability constraint

$$\overline{\{z(t)\}} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} z(t)$$

$$\overline{\{\mathbb{E}\{x_{\rightarrow d}(t)\}\}} \triangleq \sum_{l\in\mathcal{L}} \overline{\left\{\mathbb{E}\left\{x_{\rightarrow d}^{(l)}(t)\right\}\right\}} \geq \gamma \|\boldsymbol{\lambda}\|_1$$

Delivered effective packets — Reliability level × total arrival rate

System Model

- Problem Formulation

$$\mathscr{P}_1 : \min_{\boldsymbol{x}(t) \succeq 0} \overline{\{\mathbb{E}\{h(\boldsymbol{x}(t))\}\}} \qquad h(t) = \langle \boldsymbol{e}, \boldsymbol{x}(t) \rangle$$

$$\text{s.t.} \quad \overline{\{\mathbb{E}\{x_{\to d}(t)\}\}} \geq \gamma \|\boldsymbol{\lambda}\|_1$$

$$\overline{\{\mathbb{E}\{x_{ij}(t)\}\}} \leq C_{ij}, \ \forall (i,j) \in \mathcal{E}$$

$$x_{i\to}^{(l)}(t) \leq Q_i^{(l)}(t), \ \forall i \in \mathcal{V}, l \in \mathcal{L}$$

$$\text{queuing dynamics of } \boldsymbol{Q}(t)$$

- Challenges to solve the above problem

Proposed solution

- Transform it to standard form

$$\mathscr{P}_2: \min_{\boldsymbol{x}(t)\succeq 0} \overline{\{\mathbb{E}\{h(\boldsymbol{x}(t))\}\}}$$

$$\text{s.t.} \quad x_{ij}(t) \leq C_{ij}$$

stabilize the virtual queue $\boldsymbol{U}(t)$

$$U_d(t+1) = \max\{0, U_d(t) + \gamma A(t) - x_{\to d}(t)\}$$

$$U_i^{(l)}(t+1) = \max\{0, U_i^{(l)}(t) + x_{i\to}^{(\geq l)}(t) - x_{\to i}^{(\geq l+1)}(t) - a_i^{(\geq l)}(t)$$

⇕

$$\overline{\{\mathbb{E}\{x_{\to d}(t)\}\}} \geq \gamma\|\boldsymbol{\lambda}\|_1, \; \overline{\left\{\mathbb{E}\left\{x_{i\to}^{(\geq l)}(t)\right\}\right\}} \leq \overline{\left\{\mathbb{E}\left\{x_{\to i}^{(\geq l+1)}(t)\right\}\right\}} + \lambda_i^{(\geq l)}$$

Relationship (Theoretical)

- The two problems have
  - Different admissible policy space
    - Feasible set for the decision variables
  - The same network stability region
    - Set of arrival rates under which there exists at least one admissible policy
    - We present an explicit characterization for the stability region
  - The same space of network flow assignment
    - The average transmission rate for a link
    - Furthermore, the same optimal cost

Physical Interpretation

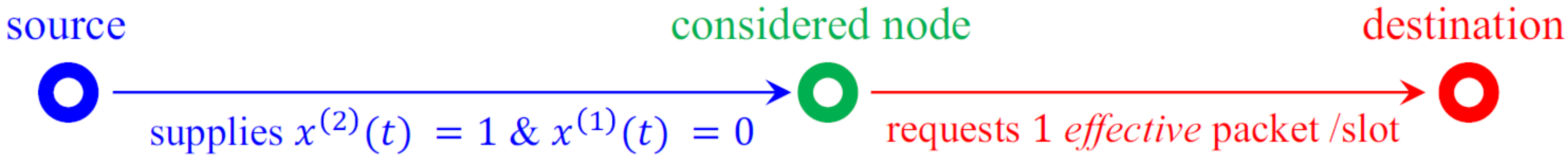
- We name the second problem virtual network
  - Imagine that each node is connected to a data-reservoir
    - The supply for packets of any lifetime is sufficient
  - Mechanism (borrow-return)
    - First borrow the packets from the reservoir to satisfy the needs
    - Then return the received packets to the reservoir
    - Virtual queue record the data deficit of the data reservoir

$$U_d(t+1) = \max\left\{0,\, U_d(t) + \gamma A(t) - x_{\to d}(t)\right\}$$

$$U_i^{(l)}(t+1) = \max\left\{0,\, U_i^{(l)}(t) + x_{i\to}^{(\geq l)}(t) - x_{\to i}^{(\geq l+1)}(t) - a_i^{(\geq l)}(t)\right.$$

Physical Interpretation

- We name the second problem virtual network
  - Equilibrium
    - Virtual queues are stabilized implies all network flows can be supported by actual packets
    - At any network location, by observing its virtual queues, we can know packets of which lifetime are available
  - Example (packets of lifetime 2 arrive at the source node)



Proposed Algorithm

- A two-step procedure
  1. Find the solution to $\mathscr{P}_2$ by Lyapunov Drift-plus-Penalty
     - Goal: min $\Delta(\boldsymbol{U}(t)) + Vh(\boldsymbol{\nu}(t)) \leq B - \langle \tilde{\boldsymbol{a}}, \boldsymbol{U}(t) \rangle - \langle \boldsymbol{w}(t), \boldsymbol{\nu}(t) \rangle$

$$w_{ij}^{(l)}(t) = -Ve_{ij} - U_i^{(\leq l)}(t) + \begin{cases} U_d(t) & j = d \\ U_j^{(\leq l-1)}(t) & j \neq d \end{cases}$$

     - Algorithm: find the best lifetime (with max weight)

$$\nu_{ij}^{(l)}(t) = C_{ij}\, \mathbb{I}\{l = l^\star, w_{ij}^{(l^\star)}(t) > 0\}$$

     - Throughput optimal & near-optimal cost performance

Proposed Algorithm

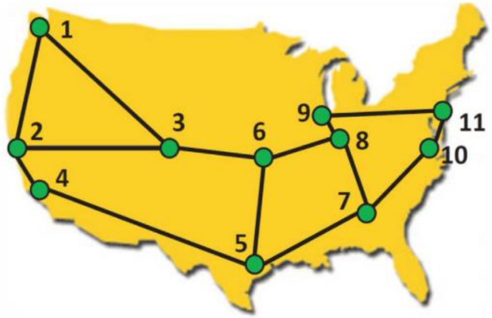
- A two-step procedure
  1. Find the solution to $\mathscr{P}_2$ by Lyapunov Drift-plus-Penalty
     - Empirical flow assignment of the above solution $\bar{\boldsymbol{\nu}}(t) = \frac{1}{t}\sum_{\tau=0}^{t-1}\boldsymbol{\nu}(\tau)$
  2. Find the solution to $\mathscr{P}_1$ based on flow matching with $\bar{\boldsymbol{\nu}}$
     - Fact a: the two problems have the same network flow assignment space
     - Fact b: given the flow assignment $\bar{\boldsymbol{\nu}}$, we can construct a randomized policy to achieve it under P1, i.e., define

$$\alpha_i^{(l)}(j) = \bar{\nu}_{ij}^{(l)} \Big/ \left(\bar{\nu}_{\to i}^{(\geq l+1)} + \lambda_i^{(\geq l)} - \bar{\nu}_{i \to}^{(\geq l+1)} \right)$$

packet of lifetime l at node i has probability $\alpha_i^{(l)}(j)$ to be sent to node j

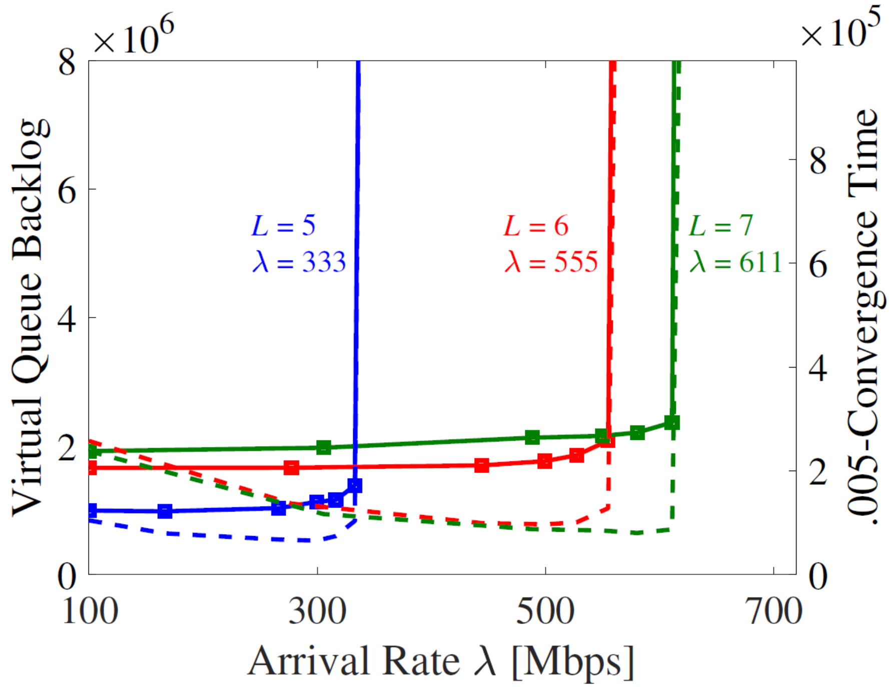
Numerical Experiments

- Configuration
  - Network topology (Abilene network)
  - Available resource & cost
    - The computational resource is 2 CPUs at any node, with cost 1 /CPU for node 5, 6, and 2 /CPU at other nodes
    - The transmission resource is 1 Gb/slot for any link, with a cost of 1 /Gb
  - Provided service
    - AgI service with 1 function: 50 Mbps/CPU, the same size of output as input
    - Two clients: (1, 9) and (3, 11)



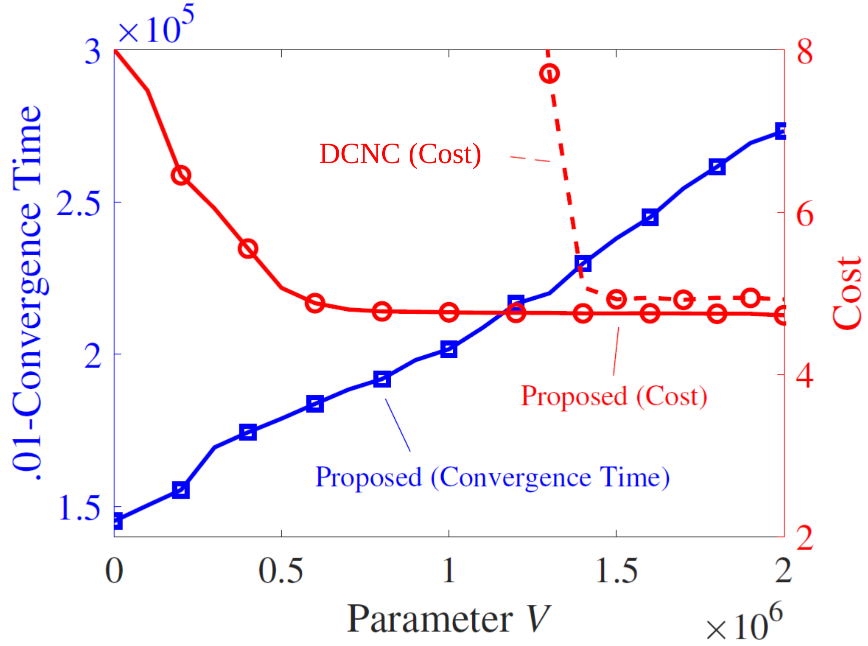
Numerical Experiments

- Network stability region
  - Actual network (solid line, convergence time), virtual network (dashed line, virtual queue backlog)
    - The same stability region
  - Effects of different lifetime



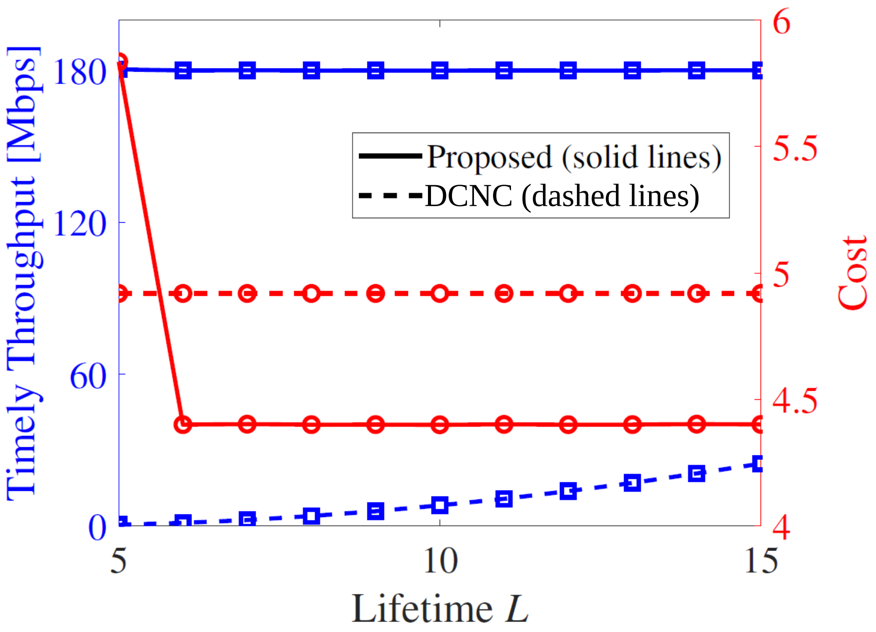
Numerical Experiments

- Tradeoff controlled by V parameter
  - $[O(V),\ O(1/V)]$ tradeoff between convergence time and the achieved cost
  - Compared to the state-of-the-art DCNC Algorithm, we attain a better cost performance
    - Drop outdated packets



Numerical Experiments

- Effects of packets' lifetime
  - Increase max-lifetime
    - DCNC: throughput grows because more packets are counted effective
    - Proposed approach: cost reduces since the data packets can detour to cheaper network locations for processing



Conclusions

- Per-packet delay is a more realistic requirement, but it is also more challenging (does not admit LDP solution!)
- The proposed approach uses virtual network to *find flow assignment*, and actual network for *routing & scheduling*
- The proposed approach significantly outperforms the DCNC algorithm in *timely throughput*

Acknowledgement

- Thanks for joining in the talk!
- Please contact yangcai@usc.edu if you have any questions, comments
- The most recent results on this topic (with peak link capacity constraint) are under review for publication at IEEE/ACM Trans. Network.