Performance and Security Challenges in Science Workflows

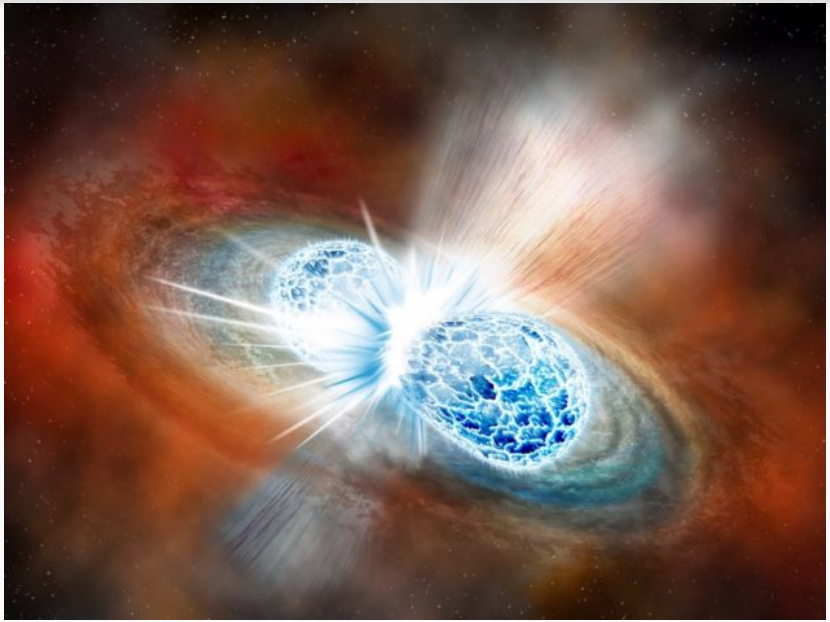
Dipak Ghosal
University of California, Davis
Lawrence Berkeley Laboratory (LBL)

SNTA 2019

Outline

- Science Workflows
- The SuperFacility Model
- Network and System Telemetry
- Deadline-Aware Flows
- Security Challenges
- Future Research

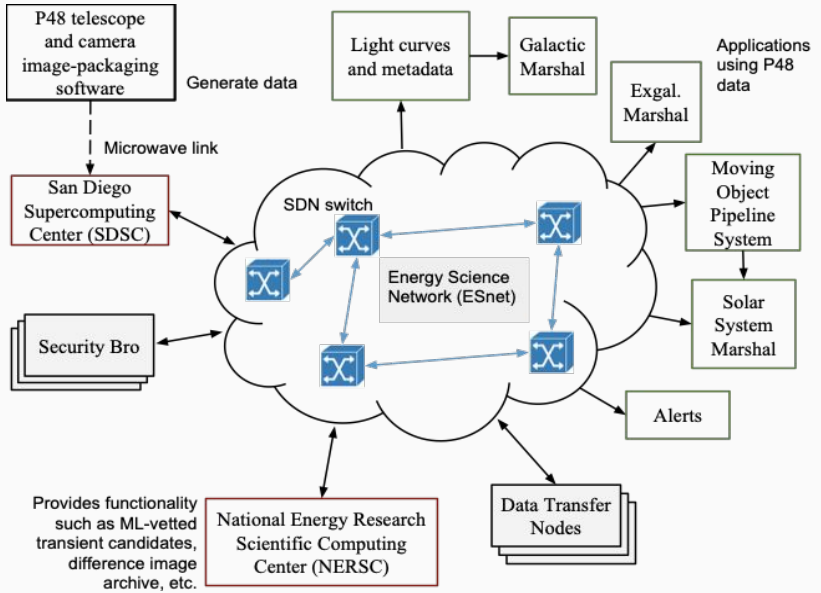
Kilonova - GW170817



Artist impression from Science.org

- Merger of two neutron stars
- It occurred in a galaxy 130 million light-years from Earth in the southern constellation of Hydra.LIGO and Virgo detectors detected the gravitational wave signal
- The data from this initial observation had to be processed in a timely manner and sent to astronomers around the world so that they could aim their instruments to the right section of the sky to image the source of the signal.

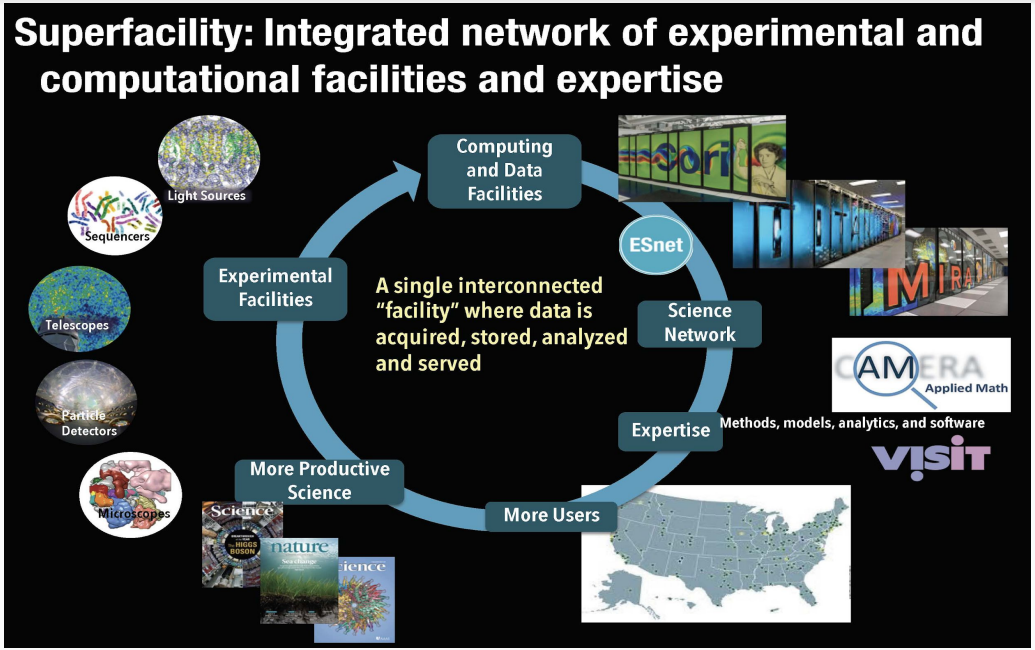
Zwicky Transient Facility (ZTF)



An abstracted version of the Zwicky Transient Facility (ZTF) workflow

- ZTF is capable of finding transients and variable stars an order of magnitude faster than the previous generation of synoptic surveys
- It generates approximately 1.3~GB uncompressed data every 45~sec
- This data is processed through multiple pipelines potentially at different networked HPC nodes to generate alerts
- Alerts must be generated within a deadline so that additional observations can be quickly scheduled during the same observation night

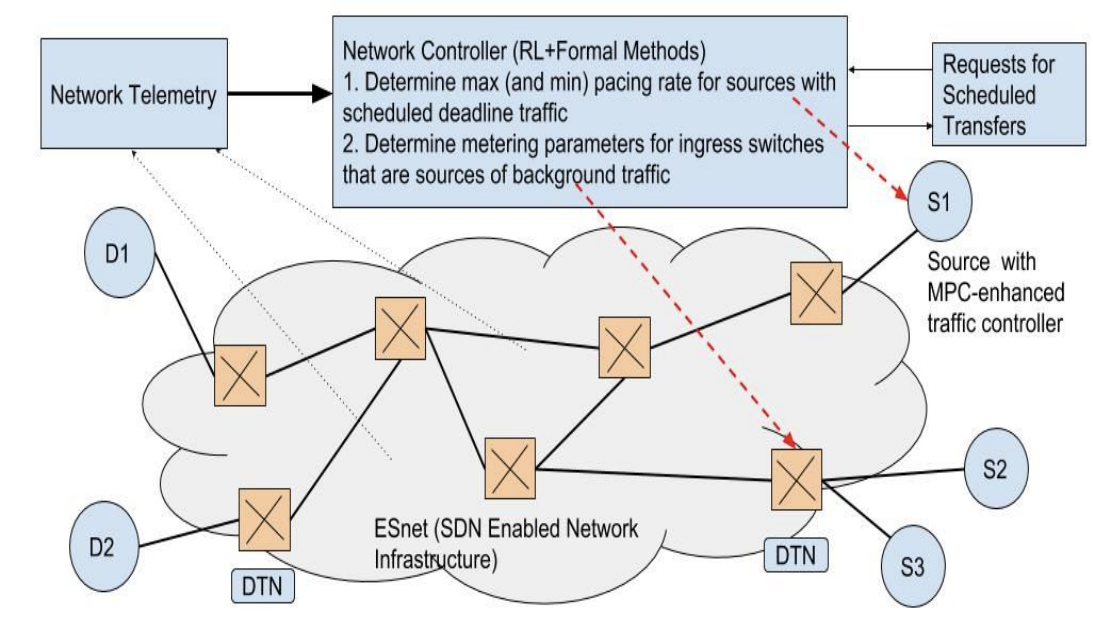
SuperFacility Model



Borrowed from Kathy Yelick's slide deck on SuperFacility Model for Data Intensive Science

1. Movement of large and complex data sets
2. Many applications must meet deadline (Real-time MRI)
3. Computational facilities need to adapt
4. The network with increased traffic - need to run network at high utilization

Abstraction



- Sources (S) and Destination (D) nodes interconnected by Software Defined Switches and Routers
- Both data for science workflows and background traffic is present
- Background traffic can be metered at the ingress router
- Data Transfer Nodes (DTN)

Network and System Telemetry

Networking Protocols

- Current transport protocols are based on the end-to-end principle
  - Network is a simple and very fast packet routing and forwarding engine
  - Intelligent and adaptable end-system
- Very minimal explicit feedback from the network
- Transport protocol make measurements to estimate network state and accordingly adapt their sending rate
- Super-successful model particularly under low to moderate network utilization

New Challenges for Science Workflows (1)

- In order to meet the traffic demands of science workflows it is essential to run the network at high utilization
  - Upwards of 90 percent utilization
- Deadline driven workflows
  - Deadline driven transfers of large data sets
  - Co-scheduling of network transfers with storage and compute resources

New Challenges for Science Workflows (2)

- Predictability
  - Small changes in requests or the network state should not result in large changes in the schedule
  - Lack of predictability will impact the utilization of the compute and storage resources
- Security
  - Denial of Service (DoS) and Distributed DoS
  - Data exfiltration and corruption through insider attack

Enabling Networking Technologies

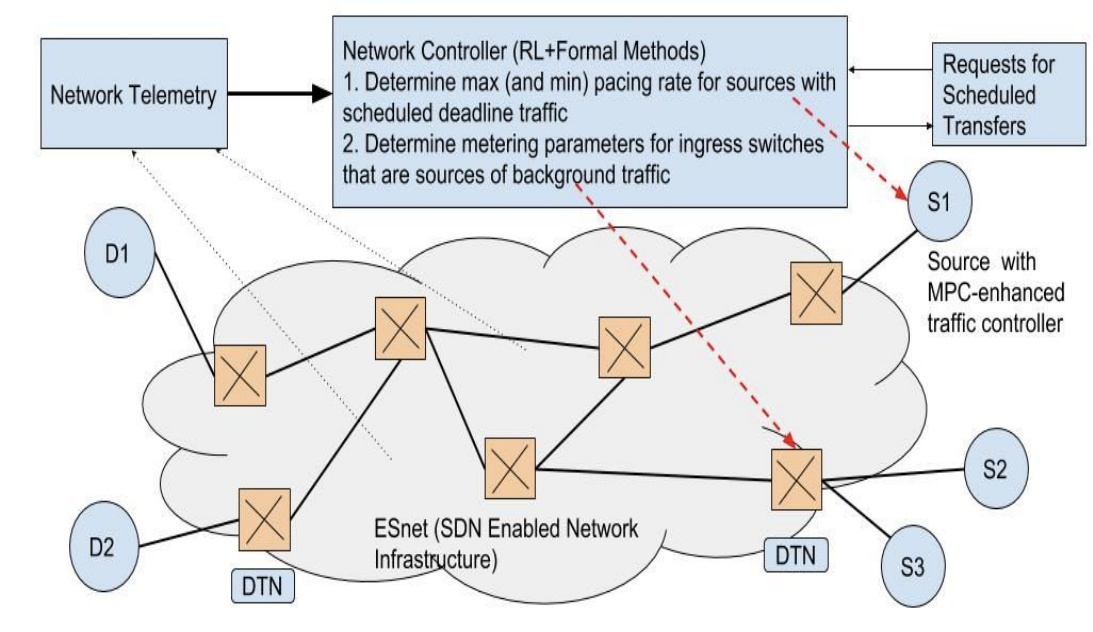
- SDN enabled softwarized networks
  - Decoupling of the control and data plane
  - Centralizing network control functions
  - Ability to program switches/routers to implement policies based on the state of the network
- Ability to pace traffic at high data rate
  - Ability to precisely control how data is injected into the network. Two broad categories: 1) Host pacing and 2) Edge pacing
  - With regards to TCP, host pacing refers to the ability to spread the transmission of the data corresponding to the allowed window over the RTT
  - Edge pacing is done at the ingress to the network
  - It is now possible to do host pacing at 40 Gbps

Enabling Technologies

- Network and System Telemetry
  - Ability to monitor and record the network state very precisely
    - Monitor queue lengths, packet delays at the network switches and routers
  - Ability to monitor and record the state of the end-system very precisely
    - Monitor the number of context switches and interrupts system wide and at per-core level
- Machine Learning
  - Traffic engineering (capacity allocation, routing)
  - **Resources allocation**
  - **Anomaly detection**

Scheduling Deadline-Aware Flows

Problem Definition



A two-level autonomous control system consisting of

1) **A Network Controller for Predictable Completion Time of Deadline-aware Flows**
2) A Model Predictive Control based Approach for Pacing Deadline Flows

Background traffic can be metered at the ingress router since the switches are SDN enabled

A Network Controller for Deadline-driven Flows

- A complex task of determining
  - if a request should be accepted,
  - how to pace different flows,
  - how to route different flows,
  - how to manage the background flows
- Simpler version of the problem (single resource bottleneck) has been studied as scheduling malleable jobs with deadline
- ML based approach will be adopted

Related Work (A Partial List)

1. Zhang, Hong, et al. "Guaranteeing deadlines for inter-data center transfers." IEEE/ACM Transactions on Networking (TON) 25.1 (2017): 579-595.

2. Srikanth Kandula, Ishai Menache, Roy Schwartz, and Spandana Raj Babbula. Calendaring for wide area networks. In ACM SIGCOMM computer communication review, volume 44, pages 515–526. ACM, 2014.

3. Sushant Jain et al. B4: Experience with a globally-deployed software defined wan. In ACM SIGCOMM Computer Communication Review, volume 43, pages 3–14. ACM, 2013.

4. S. Kandula, I. Menache, R. Schwartz, S. R. Babbula, "Calendaring for Wide Area Networks", ACM SIGCOMM Computer Communication Review, vol. 44, no. 4, pp. 515-526, 2015.

5. Alali, Fatma, et al. "Calibers: A bandwidth calendaring paradigm for science workflows." Future Generation Computer Systems 89 (2018): 736-745

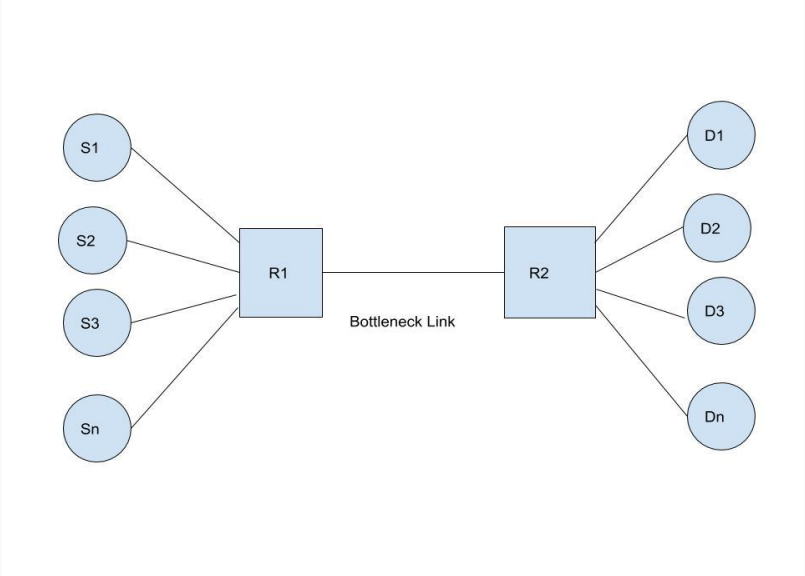
Goals

1. Can we design a Reinforcement Learning Agent that can schedule and pace flows such that the deadlines are met and network utilization is maximized?
    a. For individual flows and workflows
2. Can we design a Reinforcement Learning Agent that can meter traffic at the network ingress such that the network transfers are predictable while achieving high network utilization?
3. Can we design a Reinforcement Learning Agent that can jointly optimize routing and flow scheduling?

Model

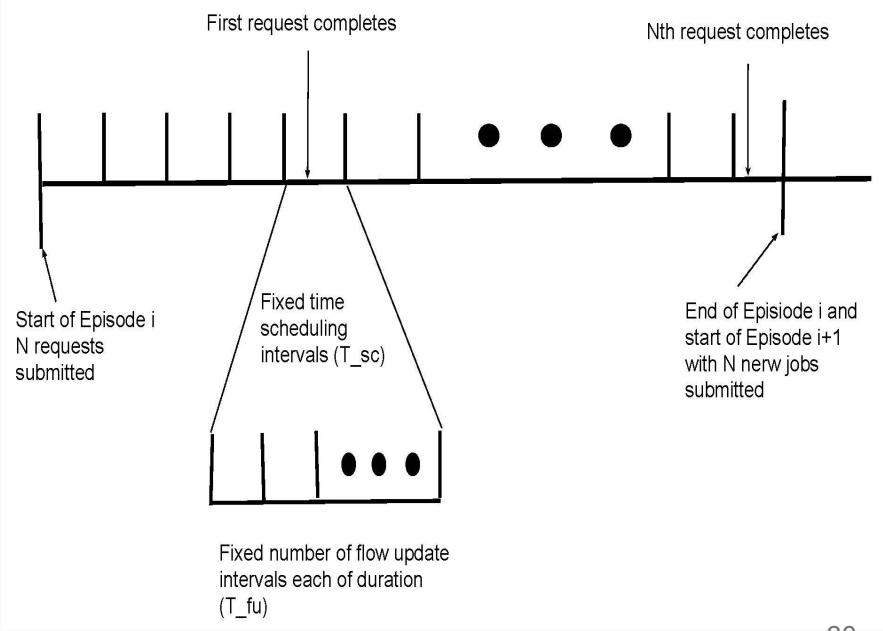
Preliminary Network

- The links from the sources to router R1 and router R2 to destination have infinite capacities (20 Gbps)
- The link between R1 and R2 is the bottleneck link (10 Gbps)
- There are only deadline driven flows
- Flows can be paced at specified integer rates (0 is allowed)



Timing

- **Scheduling Interval:** A fixed time interval when scheduling decisions are made
- **Flow Update Interval:** Each scheduling interval is divided into a fixed number flow-update interval when flow attributes are updated
- **Episode**: This corresponds to a number of scheduling intervals when the current set of transfers complete
- Notes:
  - A Request may finish anytime within an interval
  - The length of the episodes will be different since the file sizes in the requests are different



Workload (1)

- A request is a transfer of a file of size **s** drawn from Unif (smin, smax) within a given deadline **d** also drawn from Unif (dmin, dmax)
- Given a size s and a deadline d, we can define R_min (= **s/d**)
- If at each scheduling interval the flow is assigned R_min it will meet the deadline
- No interference is modeled in this preliminary study

Workload (2)

1. At the start of the episode, 3 requests are generated with
   a. A random filesize s = Unif (smin, smax)
   b. A random R_min = Unif (1, R_min-High)
   c. The deadline is then determined to be d = s/R_min
2. Example 1: If R_min_High is 3 then it should be possible to meet all the deadlines
3. Example 2: If R_min_High is 8 then in cases when the aggregate R_mins is greater than 10, then it will not be possible to meet all the deadlines

Heuristic Scheduling Algorithms

TCP - Equal Partition

1. If the sum of the rates allocated to the flows on a given link is greater than the link capacity, then flow gets a share of the capacity that is proportional to the RTT
2. In the preliminary study RTTs are assumed to be the same hence the capacity is “equally” partitioned
    a. For 3 active flows, 10 Gbps is divided into 4, 3, 3

Earliest Deadline First (EDF)

- Allocate all the capacity to the flow that has the earliest deadline
  - For single machine systems, EDF has been proved be the optimal policy (David Karger, Cliff Stein, and Joel Wein. Scheduling algorithms. CRC Handbook of Computer Science, 1997)
  - EDF has also been studied for packet scheduling in multihop networks with hard deadlines. For a single hop system, EDF has the same performance as the optimal offline algorithm when the system is underloaded
  - For tree-based multihop networks, EDF algorithm achieves the same performance as the optimal offline algorithm (Zhoujia Mao, Can Emre Koksal, and Ness B Shroff. Optimal online scheduling with arbitrary hard deadlines in multihop communication networks. IEEE/ACM Transactions on Networking (TON), 24(1):177– 189, 2016.)
- Since we consider a network with a single bottleneck link, we use EDF as the benchmark for the best achievable performance

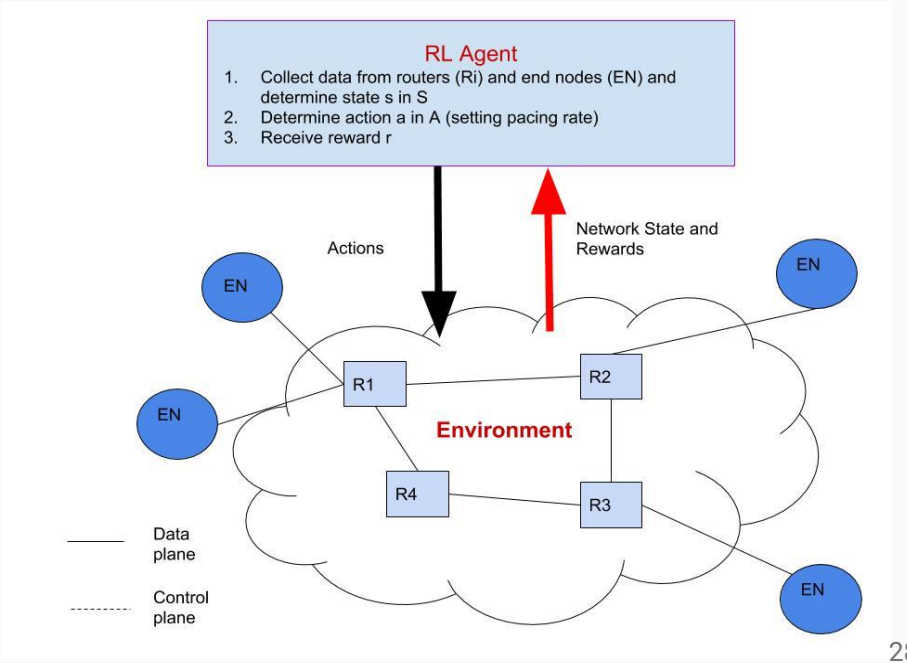
RANDOM

- In this scheme the bottleneck capacity is randomly partitioned among the number of active flows
- This is baseline case for comparison only

Reinforcement Learning

Reinforcement Learning (1)

- Policy: mapping of the perceived states of the environment to actions to be taken when in those states
- Reward Function: maps each state of the environment to a single number that indicates the desirability of the state
- Value Function: specifies what is good in the long run
- Model: The “physics” of the environment. Given the state and action it can predict the next state



Reinforcement Learning (2)

- Learning by interacting with the environment
- It is different from supervised learning
  - In interactive problems it is often impractical to obtain examples of desired behavior of all situations that the agent has to act on
- Trade-off between exploration and exploitation
  - To increase reward the agent may prefer actions that it has tried in the past but it needs to explore in order to find better action selections in the future

Application of RL in Resource Management

- Decisions are often highly repetitive (such as the periodic ZTF data)
  - Consequently, there is an abundance of training data
- RL can model complex systems and decision-making policies using deep neural networks
- RL agent can be trained for objectives that are hard-to-optimize directly since the model becomes very complex
  - Modeling
- The RL agent can operate under varying load conditions

Related Work

- Resource Management with Deep Reinforcement Learning, Hongzi Mao, Mohammad Alizadeh, Ishai Menachey, Srikanth Kandula, Massachusetts Institute of Technology & Microsoft Research, HOTNet 2016
- DeepRM is a multi-resource cluster scheduler operates in an online setting where jobs arrive dynamically and cannot be preempted once scheduled
- DeepRM learns to optimize various objectives such as minimizing average job slowdown or completion time
- DeepRM employs a standard policy gradient reinforcement learning algorithm
- DeepRM performs comparably or better than standard heuristics such as Shortest-Job-First (SJF) and a packing scheme based on Tetris

Actions

- Actions are setting the pacing rates of the flows
- We assume a discretized set of pacing rates R = {r1, r2, …., rn} from which the agent can choose the pacing rates of each flows
- For example, if the link capacity is 10Gbps, then R = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
- We assume that flows can be assigned rates (even stopped for an interval and restarted) without any penalty

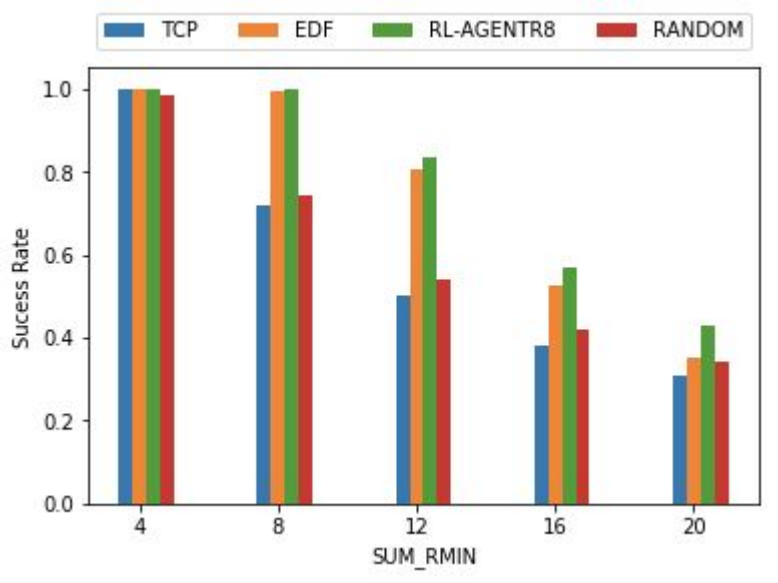
State

- The state consists of the following elements
  a. n: number of active flows
  b. b: A vector of the current R_min for each of the active flow (rounded up to nearest integer)
  c. u: utilization of the link discretized into 10 different bins { u1: 0-0.1, u2: 0.1-0.2, …., u10: > 1.0}
- Notes:
  a. In the current implementation u is always 1 since the entire bottleneck capacity is allocated
  b. A better state will be number of remaining bytes and time until deadline. This is captured in one metric - the current R_min

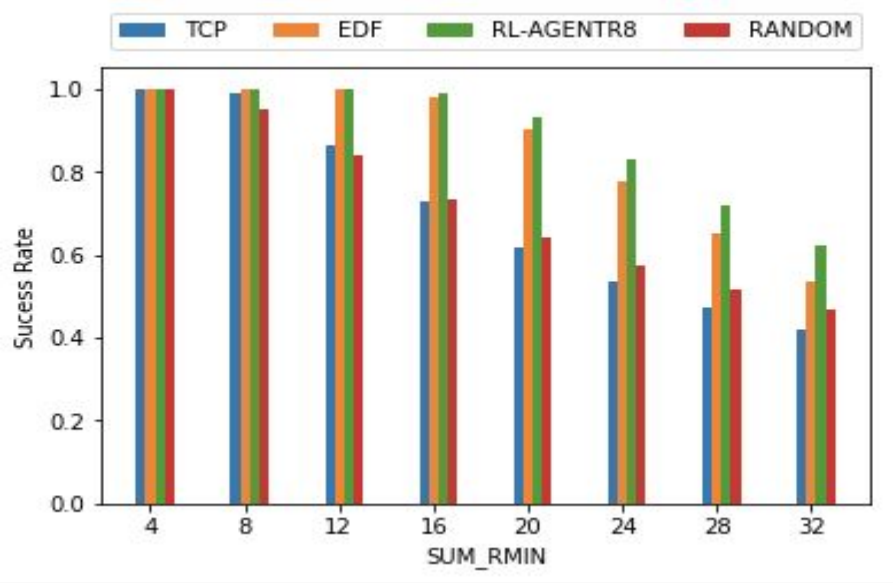
Reward Function

1. **Scheduling Interval Reward (A sample)**
   - Rmin is the minimum rate that is required for the flow to meet the deadline (filesize/deadline). At each time interval for each flow
     - a (small +1) positive reward if the rate achieved in the time slot is greater than Rmin
     - a (small -1) negative reward if the rate achieved in the time slot is less than Rmin
   - The total reward for the state action pair is the sum of the flow rewards
2. **Flow Completion Reward**
   - When a flow finishes within the given deadline a (large) positive rewards is applied with a discount factor to all the states-action pairs that helped achieve the deadline

Results

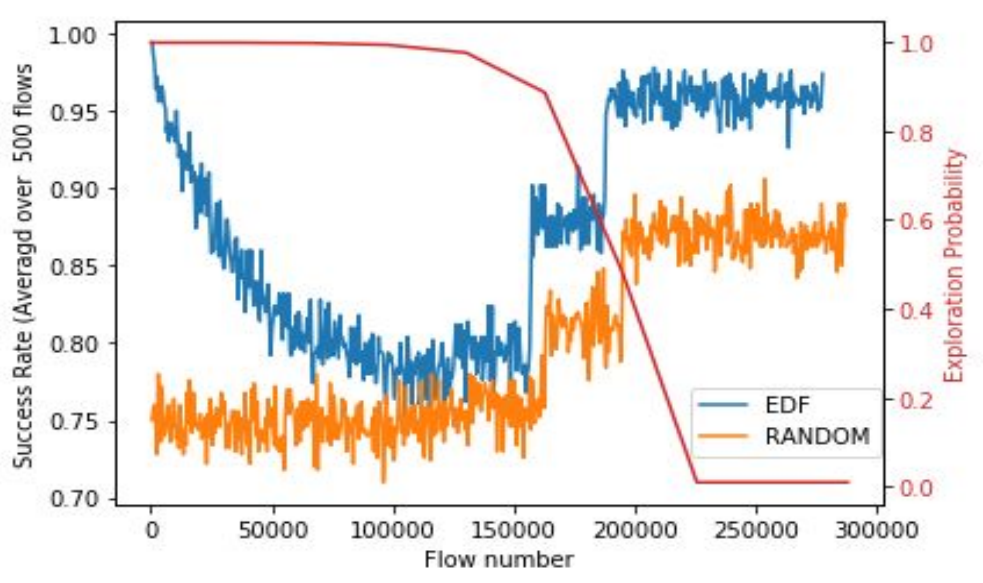


3 nodes with bottleneck link capacity of 10 Gbps



5 nodes with bottleneck link capacity of 20 Gbps

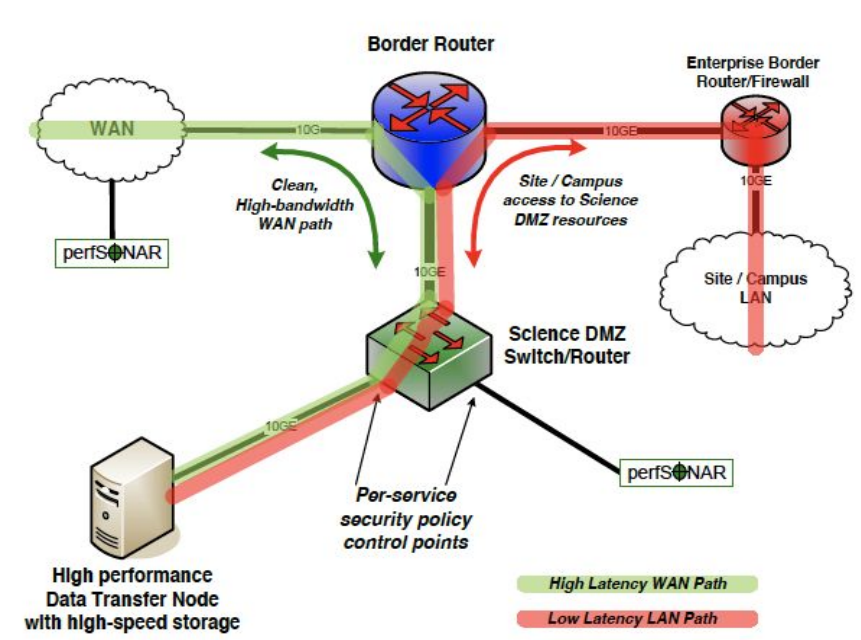
It is Really Learning?



- The red line shows how the exploration probability is changed
- The orange line shows success ratio when the new state action policy follows the Random policy
- The blue line shows the success ratio when the new state action policy follows the EDF policy

Security

Data Transfer Node (DTN)



- An important network entity in Science DMZ (E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, “The science dmz: A network design pattern for data-intensive science,” Scientific Programming, vol. 22, no. 2, pp. 173–185, 2014)
- Interface between networks of different capacities
- Interface between the HPC nodes and the network
- A performance tuned node

A Preliminary and Simple Threat Model

- External attacks
  - Denial of Service (DoS)
    - TCP SYN flood attack
    - Port Scanning attacks
- Insider attacks
  - Data exfiltration
  - Compromise data integrity

Security Challenges

- DTNs are highly performance tuned
  - Protecting the performance of such networks is an important security concern
- Science DMZs avoid typical firewalls to maximize network transfer efficiency, instead relying on various detection systems and Access Control Lists (ACLs)
- Network intrusion detection systems (NIDS), such as Bro or Snort tend to rely solely on network metrics to identify abnormal traffic or attacks
- **System performance metrics** can also reveal the type of traffic being received, including malicious traffic

End-System Performance-Based Anomaly Detection (Preliminary Results)

- Evaluated the effectiveness of system performance data in detecting TCP-SYN flood attacks on a DTN
- Hierarchical Temporal Memory (HTM) used in detection system
- System interrupts can be used to successfully detect TCP-SYN flood
    - An attack traditionally detected by network activity

Summary

- Importance of precision network and system telemetry
- Importance of Machine Learning
  - Traffic Engineering
  - Resource Allocation
  - Anomaly Detection