Augmenting Backpressure Scheduling and Routing for Wireless Computing Networks

KM Mahfujul*, Kaige Qu‡, Qiang Ye†, Ning Lu*
*Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada.
‡ Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada.
† Department of Computer Science, Memorial University of Newfoundland, St. John's, NL, Canada.
{m.kadir, ning.lu}@queensu.ca, kaige.qu@uwaterloo.ca, qiangy@mun.ca

***Abstract*—Driven by the ever-increasing computing capabilities of mobile devices, the next-generation wireless networks are evolving toward a distributed networking and computing platform, which enables in-network computing and unified resource/service provisioning. The evolution leads to a growing research interest in wireless computing networks that operate under both the high dynamics of the wireless environment and the resource heterogeneity, which complicates resource allocation, scheduling among network flows, and overall optimization. In this paper, we aim to study a low-complexity efficient solution to jointly allocate both networking resources (e.g., links to forward packets between connected computing nodes) and computing resources (e.g., computing power at each node for packet processing). We propose a novel network utility maximization problem under computing and networking resource constraints and develop an enhanced backpressure-based dynamic scheduling and routing algorithm. Finally, we verify the effectiveness of the algorithm with extensive simulations.**

I. Introduction

In evolving towards the fifth generation (5G) and beyond, wireless networks have been experiencing a paradigm shift from a pure communication network to a distributed computing platform, representing a convergence of wireless networking and computing. This is mainly due to the ever-increasing computing power of clients and devices, the emergence of machine-learning-centric applications, and the wide utilization of virtualization and service-oriented principles in various emerging edge/cloud technologies [1]. Such a convergence leads to growing research interests in *wireless computing networks*, which allow in-network computation and integrated resource/function management to realize a unified provisioning of network and computing services. The wireless computing networks are envisaged to be a key enabler for emerging applications such as event detection, real-time control and decision making, streaming data analysis, and many machine-learning-centric applications that require both fast local processing/storage and low-latency networking.

A wireless computing network can be abstracted as an undirected graph with a set of computing nodes being vertices and a set of links being edges, where each computing node with certain storage and computing capacities can host a number of service functions for data processing (such as network diminishers for compressing traffic [2] and machine learning models to process inference requests [3]) and each link with certain communication capacity can deliver data traffic over a wireless channel. The wireless computing network is to serve a set of augmented information (AgI) flows [4]. For each AgI flow, the destination node is provided with augmented information generated during the real-time processing of source data packets through a chain of multiple service functions. The service functions are hosted by computing nodes on a respective routing path, for the AgI flow.

Existing works have investigated how to jointly schedule the communication and computation resources, i.e., how to determine where to execute each service function and how to optimally route the service flows to meet service demands. The control policies for service function placements and traffic routing has been investigated in either a static or dynamic setting. For example, some studies [5]–[8] consider the static service function placement at computing nodes according to the computing/communication requirements and packet routing among the computing nodes. Since the service function placement and traffic routing are static and computing nodes are resource-limited, the packet arrival rates for the flows should be throttled (e.g, by packet dropping) in resource shortage. On the other hand, dynamic placement of service functions and traffic rerouting strategies are investigated in [4], [9], [10] in the presence of resource shortage. Although the traffic rerouting process can redistribute the computing loads across the network, it incurs a non-negligible overhead, (e.g., delay/cost of link creation and packet retransmission [10], [11]). The reconfiguration overhead also affects the scheduling and routing of other service flow packets. Therefore, we need a proper trade-off between the computing-communication resource allocation among all the service flows, for example, through the network utility maximization (NUM) framework.

Wireless computing networks require joint computing/communication resource allocation policies to satisfy the service demands of all the service flows with limited information exchange among the network components. To meet the severe changes in the service demands, it necessitates decentralized control policies that can reconfigure the service flows in resource shortage, for example, by exploring multipath-routing [12], [13]. Different from existing works, we focus on the joint computing/communication resource allocation problem while facilitating the low-complexity dynamic flow reconfiguration, which is still unexplored in the existing literature.

In this paper, we consider an NUM problem for AgI

flows in a wireless computing network through multiple service function chains. We model the proposed wireless computing network with a set of generic computing nodes of limited capacity to compute service functions via virtualization and wireless communication channels to route the data traffic among computing nodes. We address the joint computing/communication resource allocation problem in the wireless computing network by designing a dynamic packet scheduling mechanism. Specifically, we update the packet *state* after completing a functional processing step and promoting it to the subsequent downstream function residing in the same computing node, denoted as internal packet forwarding. We route the unprocessed packets to a neighboring computing node, denoted as external packet forwarding, resulting in a dynamic service flow reconfiguration in resource shortage.

The computing nodes backlog the packets based on their current *state* and operate these packet-forwarding strategies independently across the network with limited information exchange among neighboring computing nodes. Augmenting these packet-forwarding strategies offer a low-complexity decentralized control policy for a joint computing/communication resource allocation problem. Moreover, the *state*-dependent packet forwarding scheme does not incur any reconfiguration delay since internal/external forwarding of some arbitrary packets does not affect any other flow packets. We formulate a utility maximization problem as a function of the service flow packet admission rate to guarantee optimal resource utilization. By utilizing convex duality, the proposed NUM formulation develops an augmented backpressure-based decentralized scheduling and routing algorithm. Finally, we provide extensive simulations to cross-verify the algorithm performance.

The rest of this paper is organized as follows. In Section II, we introduce the network model and the NUM problem formulation. In Section III, we elaborate the design of the dynamic backpressure-based scheduling and routing algorithm by decoupling the problem into solvable sub-problems with solutions. We verify our proposed solutions by simulations in Section IV and we conclude our paper and discuss the future works in Section V.

II. System Model and Problem Formulation

A. Network Model and Packet States

We consider a wireless computing network which consists of computing nodes in set $\mathcal{N}$ and wireless communication channels. For each service flow, packets arrive at a fixed source node and packets leave the network at a fixed destination node. The computing nodes can both process and forward the packets from a set of service flows $\mathcal{R}$.

Packets from a flow $r \in \mathcal{R}$ requires a chain of network/computing functions, distributed across the network. Let packets from flow r be sequentially processed by $|K^{(r)}|$ network/computing functions. As each function processes the packets and may update the packets by revising the packet header/payload, we associate $|K^{(r)}|$ functions with $|K^{(r)}|$ packet states. When a packet finishes processing by function

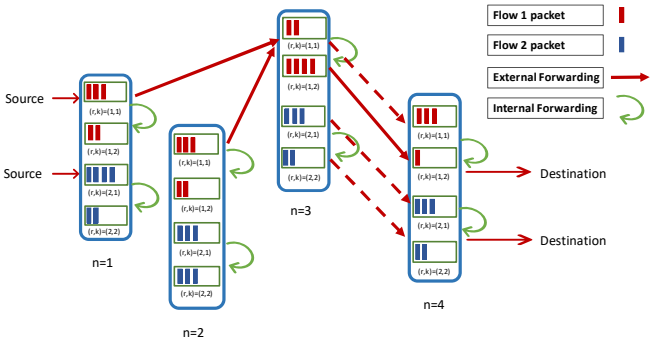


Fig. 1: Wireless Computing Network: A wireless computing network with four nodes, $|R| = 2$ service flows with $|K| = 2$ states is shown. Node $n = 1$ is the source node, and $n = 4$ is the destination of the computing network. From $n = 3$ to $n = 4$, there are four candidate queues to externally forward their packets, the algorithm may choose $(r, k) = (1, 2)$ to forward its packets at current slot.

$k \in \mathcal{K}$, meaning the packet completes a specific type of processing, the packet state k is updated, and the packet will be subsequently processed by downstream functions in the chain. Here, we index the functions and states analogously by k.

Each computing node $n \in \mathcal{N}$ can host $|K^{(r)}|$ functions for each flow $r \in \mathcal{R}$. To support such processing, node n maintains $|K^{(r)}|$ virtual processing queues for each flow r, indexed by (r, k) pairs, and there are $R \times K$ virtual queues at node n in total. Details are illustrated in Fig. 1.

B. Flow Model

Flow r has an exogenous packet arrival process. Assume that the packet arrival is i.i.d. over time with finite rate $\lambda^{(r)} \cdot \mathbf{1}_{(k=1,n=src)}$, denoting that the packet arrival/source rate of flow r at the source node $n = src$ and the packet is at its earliest state $k = 1$. [1] The arrival rate is upper-bounded by $\lambda^{(r)} \leq \lambda_r^{max} < \infty$. Additionally, we consider a system with saturated traffic where the source node can always receive packets at any time slot [14].

A computing node $n \in \mathcal{N}$ processes packets from queue $Q_n^{(r,k-1)}(t)$ (i.e., packets of flow r currently on state $k-1$, in a queue indexed by $(r, k-1)$) at time slot t with a processing rate $a_n^{(r,k-1)}$ and promotes to queue $Q_n^{(r,k)}(t)$ at the same node by the end of slot t. We denote this process as internal packet forwarding. When the computation load at the node becomes high, the node can migrate some (r, k) packets to a neighboring node $j \in \mathcal{N}$ at rate $s_{n,j}^{(r,k)}$, denoted as external packet forwarding. The migrated packets are either stored in the queue $Q_j^{(r,k)}(t)$, for processing in the future time slots, or externally forwarded to another node. Considering all the

[1] where $\mathbf{1}$ is representing an indicator function defined as

$$\mathbf{1}_{(k=1,n=src)} = \begin{cases} 1 & if \quad (k = 1 \text{ and } n = src), \\ 0 & \text{otherwise.} \end{cases}$$

variables, we design a flow conservation constraint for each (r,k), given by,

$$\lambda^{(r)} \cdot \mathbf{1}_{(k=1,n=src)} + \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)} + a_n^{(r,k-1)} \leq \sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)} + a_n^{(r,k)}, \quad \forall n \in \mathcal{N}, \quad \forall r \in \mathcal{R}, \quad \forall k \in \mathcal{K}, \tag{1}$$

where node n accepts the packet arrivals only at the source node with packet state $k = 1$, receives internal packet forwarding from the same flow $(r, k-1)$ at rate $a_n^{(r,k-1)}$, and receives external packet forwarding from the upstream (e.g., source node) at rate $s_{i,n}^{(r,k)}$. Similarly, the node can promote the packets to the next state (e.g., from $k-1$ to k) at rate $a_n^{(r,k)}$ and internally forwards to the virtual queue $Q_n^{(r,k+1)}(t)$, by the end of slot t. The node can also externally forward some packets to the downstream (e.g., toward destination) at rate $s_{n,j}^{(r,k)}$ to balance the computation load. The inequality sign is because the total arrival rates should be less than or equal to the total departure rates. We design the associated virtual queue for every (r,k) pair as

$$Q_n^{(r,k)}(t+1) = \left[Q_n^{(r,k)}(t) - \sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)}(t) - a_n^{(r,k)}(t)\right]^+ + \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)}(t) + a_n^{(r,k-1)}(t) + \lambda^{(r)} \cdot \mathbf{1}_{(k=1,n=src)}(t), \tag{2}$$

where the virtual queues will create bounded actual data queues in the network, faithfully implementing the resulting algorithm with the actual data queues.

1) Computing Resource: Each node has limited CPU processing power in CPU cycles/slot, denoted by $\mathcal{C}_n$, for node n. The functions may require different CPU cycles to process a single bit of flow packets, denoted by $\rho^{(r,k)}$ for flow r at state k. Additionally, we denote the flow packet size by $\sigma^{(r)}$ in bits, which may vary according to flow type r. The total number of required CPU cycles for processing function k with rate $a_n^{(r,k)}$ should not exceed the available resource, i.e.,

$$a_n^{(r,k)} \cdot \sigma^{(r)} \rho^{(r,k)} \leq \mathcal{C}_n. \tag{3}$$

2) Communication Resource: When the computation load in the computing nodes become high, the nodes externally forward the flow packets to the downstream via the communication channels. The channel states are time-varying. Node $n \in \mathcal{N}$ can forward packets to at most one downstream node $j \in \mathcal{N}$ at a given time slot, i.e.,

$$\sum_{j\in\mathcal{N}} \sum_{(r,k)\in(\mathcal{R},\mathcal{K})} z_{n,j}^{(r,k)}(t) \leq 1, \quad \forall n \in \mathcal{N}. \tag{4}$$

The packets are externally forwarded over the link (n,j), to the downstream at a rate $s_{n,j}^{(r,k)}$, denoting the actual packet forwarding rate. We provide the relationship with potential packet forwarding rate (the maximum allowable rate facilitated by the link) $\mu_{n,j}$ by,

$$s_{n,j}^{(r,k)} \leq z_{n,j}^{(r,k)}(t) \cdot \mu_{n,j}. \tag{5}$$

C. Problem Formulation

To maximize the total packet admission rate of all flows $r \in \mathcal{R}$, we use a network utility function for each flow r as a concave increasing function of packet admission rate denoted as $U_r(\lambda^{(r)})$. We define $U_r(\cdot)$ as a concave differentiable function with at least a continuous first derivative, upper-bounded by $U_r^{'}(\cdot) \leq \eta^{(max)}$. Considering all the network constraints, we formally present the NUM problem:

$$\textbf{(P1)} \quad \max_{\boldsymbol{\lambda^{(r)}}, \boldsymbol{a_n^{(r,k)}}, \boldsymbol{s_{m,n}^{(r,k)}}} \sum_{r=1}^{R} U_r(\lambda^{(r)}) \tag{6}$$
$$\text{subject to} \quad (1),(3),(4),(5),$$
$$\forall n \in \mathcal{N}, \quad \forall r \in \mathcal{R}, \quad \mathcal{K} = [1, K^{(r)}],$$
$$\forall k \in \mathcal{K}, k-1 \geq 1, k \leq K^{(r)}.$$

III. Dynamic Backpressure-based Framework

The utility optimization problem **(P1)** is a convex optimization problem subject to linear constraints. For simplicity, we introduce a function $h(\cdot)$ of $y_n^{(r,k)}$ to capture the flow conservation constraint in Eq. (1), given by,

$$h(y_n^{(r,k)}) = \lambda^{(r)} \cdot \mathbf{1}_{(k=1,n=src)} + \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)} + a_n^{(r,k-1)} - \sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)} - a_n^{(r,k)}. \tag{7}$$

Therefore, constraint (1) can be rewritten as

$$h(y_n^{(r,k)}) \leq 0. \tag{8}$$

Further, we simplify the sum of utility function in Eq. (6) as $\sum_{r=1}^{R} f_r$ and rewrite utility maximization problem as

$$\max_{\boldsymbol{h}} \sum_{r=1}^{R} f_r \tag{9}$$
$$s.t. \quad h(y_n^{(r,k)}) \leq 0, \quad \forall n \in \mathcal{N}, \tag{10}$$
$$(3),(4),(5).$$

In order to solve this non-linear convex problem, we introduce the Lagrange multiplier $\theta_n^{(r,k)} \in \mathbb{R}^{\mathcal{R}\times\mathcal{K}}$. The Lagrange dual problem is written as

$$\mathcal{L}(\cdot) = \sum_{r=1}^{R} f_r - \sum_{\substack{(r,k)\in(\mathcal{R},\mathcal{K}),\\ n\in\mathcal{N}}} \theta_n^{(r,k)} \cdot h(y_n^{(r,k)}) \tag{11}$$
$$s.t. \quad y_n \in \mathcal{A},$$

where we simplify the constraints by letting $(3),(4),(5) = \mathcal{A}$, (i.e., capacity region). The lagrangian dual problem can be decoupled into sub-problems based on the decision variables.

$$\textbf{(SP1)} \quad \max \sum_{\substack{(r,k)\in(\mathcal{R},\mathcal{K}=1),\\ n=src}} \left[f_r - \theta_n^{(r,k)} \cdot h(y_n^{(r,k)})\right] \tag{12}$$

In **(SP1)**, we consider $n = src, k = 1$, and the exogenous arrival/source rate $\lambda^{(r)}$.

In other cases, when $n \neq src$ and $k \neq 1$, then the lagrangian in Eq. (11) includes the variables $s_{n,j}^{(r,k)}$ and $a_n^{(r,k)}$, formulating joint scheduling and routing problem as

$$\begin{aligned}\textbf{(SP2)} \quad & \max \quad - \sum_{\substack{(r,k)\in(\mathcal{R},\mathcal{K}\setminus 1),\\ n\in\mathcal{N}\setminus src}} \theta_n^{(r,k)} \cdot h(y_n^{(r,k)}) \\ & = \max \quad \sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)}(t) - \theta_n^{(r,k+1)}\right] \\ & \quad + \sum_{\substack{(r,k),\\ n}} s_{n,j}^{(r,k)} \sum_{j\in\mathcal{N}} \left[\theta_{n,j}^{(r,k)} - \theta_{j,n}^{(r,k)}\right]. \end{aligned} \tag{13}$$

Proof. Appendix A. □

We simplify the summation $\sum_{\substack{(r,k)\in(\mathcal{R},\mathcal{K}\setminus 1),\\ n\in\mathcal{N}\setminus src}}(\cdot) = \sum_{\substack{(r,k),\\ n}}(\cdot)$. We can solve the resource allocation problem in **(SP2)** if we can estimate the value of Lagrange multiplier (i.e., $\theta_n^{(r,k)}$) from Eq. (13). Updating $\theta_n^{(r,k)}$ according to a virtual queue dynamics (i.e., as $\theta_n^{(r,k)}(t) = Q_n^{(r,k)}(t)$) from Eq. (2), provides us an estimation. The details can be found in [15]. According to our analysis presented by Eq. (12), and Eq. (13), we can provide the deterministic solutions to problem **(P1)** by decoupling it into sub-problems with readily available solutions. The sub-problems are solved sequentially and together they provide the joint algorithm to solve initial problem **(P1)** as follows.

*1) **(SP1): Flow Control**:*

$$\max_{\boldsymbol{\lambda^{(r)}}} \sum_{\substack{(r,k=1),\\ n=src}} \left[V.U_r(\lambda^{(r)}) - Q_n^{(r,k)}(t)\cdot\lambda^{(r)}\right] \tag{14}$$
$$s.t. \quad \lambda^{(r)} \in dom(U_r)$$

Remark 1. *Since exogenous flows are independent,* $\max\sum(\cdot) = \sum\max(\cdot)$ *in **SP1**.*

The solution to **(SP1)** is given by,

$$\lambda^{(r^*)} = \left[U'^{-1}\left(\frac{Q_n^{(r,k)}(t)}{V}\right)\right]_0^{\lambda_r^{max}} \tag{15}$$

We decouple the problem **(SP2)** into two sub-problems.

*2) **(SP2A): Internal Forwarding**:*

$$\max_{\boldsymbol{a_n^{(r,k)}}} \sum_{\substack{(r,k:k+1\leq K^{(r)}),\\ n\in\mathcal{N}}} a_n^{(r,k)}\left[Q_n^{(r,k)}(t) - Q_n^{(r,k+1)}(t)\right] \tag{16}$$
$$s.t. \quad (3) \text{ and } a_n^{(r,k)} \geq 0$$

The solution to this sub problem is given by the following,

$$\begin{aligned} & a_n^{(r,k*)} \\ & = argmax_{\boldsymbol{(r,k)}}\left[Q_n^{(r,k)}(t) - Q_n^{(r,k+1)}(t)\right]\cdot\frac{\mathcal{C}_n}{\sigma^{(r)}\rho^{(r,k)}} \end{aligned} \tag{17}$$

*3) **(SP2B): External Forwarding**:*

$$\max_{\boldsymbol{s_{n,j}^{(r,k)}}} \sum_{(r,k)\in(\mathcal{R},\mathcal{K}\leq K^{(r)})} s_{n,j}^{(r,k)} \sum_{j\in\mathcal{N}}\left[Q_{n,j}^{(r,k)}(t) - Q_{j,n}^{(r,k)}(t)\right] \tag{18}$$
$$s.t. \quad (4), (5) \text{ and } s_{n,j}^{(r,k)} \geq 0.$$

we incorporate a maximum backpressure-based scheduling between the virtual queues (in between the nodes to downstream) and available transmission rates at the given time slot.

$$\begin{aligned} s_{n,j}^{(r^*,k^*)} = argmax_{\boldsymbol{\mu_{n,j}}} \sum_{n,j\in\mathcal{L}} \mu_{n,j}\cdot w_{n,j}^{(r^*,k^*)}(t), \\ \forall j \in Neigh(n), \end{aligned} \tag{19}$$

where,

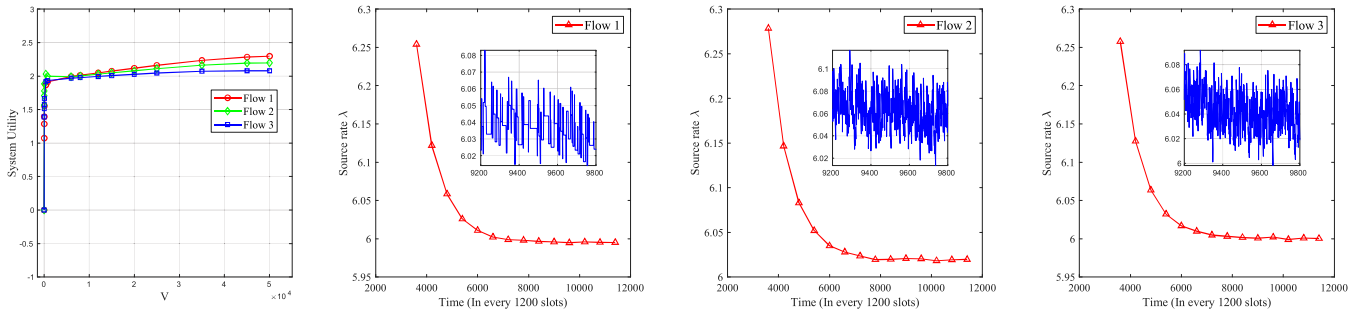
$$\begin{aligned} w_{n,j}^{(r^*,k^*)}(t) = argmax_{\boldsymbol{(r,k)}}[Q_{n,j}^{(r,k)}(t) - Q_{j,n}^{(r,k)}(t)], \\ \forall j \in Neigh(n), \forall(r,k) \end{aligned} \tag{20}$$

The achievable rate $\mu_{n,j}$ is given by the matrix $\boldsymbol{\mu}$ at each slot. It is important to note that, the actual rate is given by, $s_{n,j}^{(r^*,k^*)} = min[\mu_{n,j}, Q_{n,j}^{(r,k)}(t) - Q_{j,n}^{(r,k)}(t)]$, and when $[Q_{n,j}^{(r,k^*)}(t) - Q_{j,n}^{(r,k^*)}(t)] \leq 0$, we set $\mu_{n,j} = 0$.

IV. Experimental Evaluation

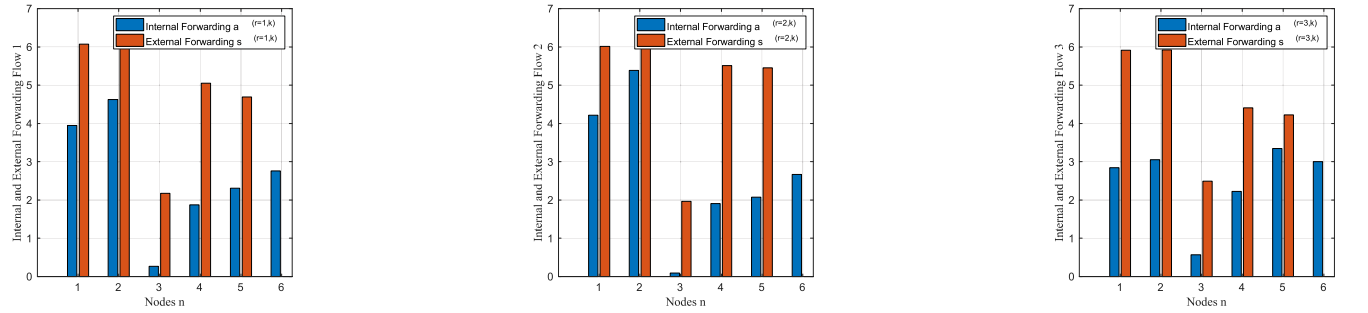
We evaluate the performance of the proposed algorithm by performing extensive simulations. First, we briefly discuss the simulation settings and then we present our experimental results. We control the exogenous packet arrival/source rate at the source node of the network. Fig. 2a shows the achievable utility by accommodating packet arrivals from 3 different flows in the network under different value of the scaling parameter V, according to the Eq. (15). We use $log(1+\lambda^{(r)})$ as our system utility metric. In the experiment, we fixed maximum achievable packet arrival/source rates of *flow1-flow3* as $\lambda_r^{max} = [9, 8, 7]$. We also set $\lambda^{(r^*)} = 0$, when $V = 0$ according to Eq. (15). The source rate $\lambda^{(r^*)}$ increases very sharply with the increment of V, with large value, (e.g., $V = 40000, 50000$) the source rates converge to the maximum achievable rates.

The proposed algorithm stabilizes the network utility to a level while maintaining fairness among the different flows. We experiment on the convergence of network utilities over $T = 12000$ time slots with a fixed $V = 20000$. We compute the time-average flow rates during $t = 600$ time slots, showing the changes in achievable utility during small time windows. The source queues of the networks are initially empty. Thus, data queues can accept more packets from the network reservoirs. The parameter V regulates the number of flow packets that a source queue can accept. When V is very large, source queues, i.e., $Q_{n=src}^{(r,k=1)}(t)$, can allow more flow packets and add them to the current backlog. Therefore, with sufficiently large enough V, a source queue can accept the number of packets to its maximum limit, i.e., λ_r^{max}.



(a) System utility vs. V parameters. *(b) Source rate of Flow1.* *(c) Source rate of Flow2.* *(d) Source rate of Flow3.*

Fig. 2: Convergence of flow packet admissions.



(a) Internal vs. External forwarding rates in all nodes of Flow 1. *(b) Internal vs. External forwarding rates in all nodes of Flow 2.* *(c) Internal vs. External forwarding rates in all nodes of Flow 3*

Fig. 3: The comparison between Internal forwarding and External Forwarding rates in each computing nodes of the network.

With time progression, the queue backlogs become large since the system can accommodate limited internal/external packet forwarding due to resource limitations. Therefore to avoid congestion and instability, the source nodes regulate the admission of flow packets over time. As a result, the admission/source rates in the network decrease over time until the rates converge to stability. The simulations presented in Fig. 2b-2d demonstrate the aforementioned trends in admission rates. Along with the time-average rate, we present a detailed speculation of the utility dynamics over each time slot (e.g., $t = [9200 - 9800]$). Utility dynamics in the region show that source rates do not fluctuate significantly, varying at most at a range of 0.10.

We perform a comparative experiment on internal/external packet forwarding rates at each computing node $n = [1, 6]$ presented in Fig. 3. We fix the computing resource at each node by some arbitrary rate (e.g., $[145, 175, 160, 130, 175, 140]$ $cycles/slot$). We set the packet size of each flow by $\sigma^{(r)} = [3, 3.5, 3] bits$. The flow packets require a chain of network/computing function processing distributed across the network. The functions are specific to the flow type, and their computing processing loads to process a packet vary. We assume that each of the flows requires processing by three functions. For example, the functions of *flow1* require $\rho^{(r=1,k=1\rightarrow 3)} = [2, 3, 4] cycles/bits$ to process. The internal forwarding rate of packets depends on the computational load at the node. For example, flow r packets currently waiting in the queue $Q_n^{(r,k)}$ at node n, will be promoted to $Q_n^{(r,k+1)}$ according to Eq. (17) by performing a state update. The node can externally forward the packets according to Eq. (19) when the computational load becomes high. The potential external forwarding rate $\mu_{n,j}$ is given by the matrix $\boldsymbol{\mu}$ (e.g., $\mu_{n,j} = 12 \rightarrow 19$ $packets/slot$ if a directed time-varying wireless channel exists between nodes $n - j$, otherwise $\mu_{n,j} = 0$, and $\mu_{n,n} = 0$). The internal/external forwarding rates do not directly depend on the parameter V. The rate evolves depending on computing/communication resources and current queue backlogs. Fig. 3 shows the internal/forwarding rate at each node. A node makes the best usage of its computing/communication resources and at a time slot by choosing internal/external forwarding of packets with a rate given by the proposed algorithm.

V. Conclusion

We studied a dynamic scheduling and routing problem in the wireless computing network via packet forwarding strategies under an NUM framework. The strategies involve efficient resource allocation decisions with minimum information exchange that develops an augmented backpressure-based algorithm. In the future, we will consider to study the storage resource allocation with scheduling and routing in AgI service flows under mobility constraints.

REFERENCES

[1] Q. Duan, S. Wang, and N. Ansari, "Convergence of networking and cloud/edge computing: Status, challenges, and opportunities," *IEEE Network*, vol. 34, no. 6, pp. 148–155, 2020.

[2] Y. Chen, J. Wu, and B. Ji, "Optimizing flow bandwidth consumption with traffic-diminishing middlebox placement," in *Proc. of 49th International Conference on Parallel Processing-ICPP*, virtual event, 2020.

[3] M. Shi, X. Lin, and L. Jiao, "Power-of-2-arms for bandit learning with switching costs," in *Proc. of ACM MobiHoc*, Seoul, South Korea, 2022.

[4] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal control of wireless computing networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 12, pp. 8283–8298, 2018.

[5] M. Barcelo, J. Llorca, A. M. Tulino, and N. Raman, "The cloud service distribution problem in distributed cloud networks," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 344–350.

[6] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *Proc. of IEEE INFOCOM*, Atlanta, GA, USA, 2017.

[7] J. Li, W. Shi, Q. Ye, S. Zhang, W. Zhuang, and X. Shen, "Joint virtual network topology design and embedding for cybertwin-enabled 6G core networks," *IEEE Internet of Things Journal*, vol. 8, no. 22, pp. 16 313–16 325, 2021.

[8] L. Gu, D. Zeng, S. Tao, S. Guo, H. Jin, A. Y. Zomaya, and W. Zhuang, "Fairness-aware dynamic rate control and flow scheduling for network utility maximization in network service chain," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1059–1071, 2019.

[9] J. Zhang, A. Sinha, J. Llorca, A. M. Tulino, and E. Modiano, "Optimal control of distributed computing networks with mixed-cast traffic flows," *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1760–1773, 2021.

[10] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2394–2408, 2020.

[11] C.-H. Wang, J. Llorca, A. M. Tulino, and T. Javidi, "Dynamic cloud network control under reconfiguration delay and cost," *IEEE/ACM Transactions on Networking*, vol. 27, no. 2, pp. 491–504, 2019.

[12] Q. Liu, H. Zeng, and M. Chen, "Network utility maximization under maximum delay constraints and throughput requirements," *IEEE/ACM Transactions on Networking*, vol. 28, no. 5, pp. 2132–2145, 2020.

[13] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, 2006.

[14] L. Georgiadis, M. J. Neely, L. Tassiulas *et al.*, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends® in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[15] R. Srikant and L. Ying, *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[16] A. Eryilmaz and R. Srikant, "Joint congestion control, routing, and MAC for stability and fairness in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1514–1524, 2006.

[17] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 2, 2004, pp. 1484–1489 Vol.2.

APPENDIX A

According to Eq. (13), we maximize a negative term:

$$max \quad - \sum_{\substack{(r,k)\in(\mathcal{R},\mathcal{K}\setminus K^{(r)}),\\ n\in\mathcal{N}\setminus Dst}} \theta_n^{(r,k)} \cdot h(y_n^{(r,k)})$$

where, $h(y_n^{(r,k)})$ is given by Eq. (7) and we consider the internal and external packet forwarding.

$$= \sum_{\substack{(r,k),\\ n}} \theta_n^{(r,k)} \left[\sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)} - a_n^{(r,k)} - \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)} + a_n^{(r,k-1)} \right]$$

$$= \sum_{\substack{(r,k),\\ n}} \theta_n^{(r,k)} \left[a_n^{(r,k)} - a_n^{(r,k-1)} \right] + \sum_{\substack{(r,k),\\ n}} \theta_n^{(r,k)} \left[\sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)} - \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)} \right].$$

We rewrite the above terms by performing trivial algebraic manipulations [15]–[17], also it is important to note that, we can rewrite the first term as, $\left[\theta_n^{(r,k)} - \theta_n^{(r,k-1)}\right] = \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)}\right]$, since we are collecting the difference between two adjacent lagrange multipliers, which in practice can be modeled as the backpressure difference between two queues. Therefore, with a little change in index-terms, i.e., $(r, k+1)$ such that $k+1 \leq K^{(r)}$, we collect the backpressure difference:

$$\sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)} \right] + \left[\sum_{\substack{(r,k),\\ n}} \theta_n^{(r,k)} \sum_{j\in\mathcal{N}} s_{n,j}^{(r,k)} - \sum_{\substack{(r,k),\\ n}} \theta_n^{(r,k)} \sum_{i\in\mathcal{N}} s_{i,n}^{(r,k)} \right]$$

$$= \sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)} \right] + \left[\sum_{\substack{(r,k),\\ n}} \sum_{j\in\mathcal{N}} \theta_{n,j}^{(r,k)} s_{n,j}^{(r,k)} - \sum_{\substack{(r,k),\\ n}} \sum_{j\in\mathcal{N}} \theta_{j,n}^{(r,k)} s_{j,n}^{(r,k)} \right]$$

$$= \sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)} \right] + \left[\sum_{\substack{(r,k),\\ n}} \sum_{j\in\mathcal{N}} \theta_{n,j}^{(r,k)} s_{n,j}^{(r,k)} - \sum_{\substack{(r,k),\\ n}} \sum_{j\in\mathcal{N}} \theta_{j,n}^{(r,k)} s_{n,j}^{(r,k)} \right]$$

$$= \sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)} \right] + \sum_{\substack{(r,k),\\ n}} s_{n,j}^{(r,k)} \left[\sum_{j\in\mathcal{N}} \theta_{n,j}^{(r,k)} - \sum_{j\in\mathcal{N}} \theta_{j,n}^{(r,k)} \right]$$

$$= \sum_{\substack{(r,k),\\ n}} a_n^{(r,k)} \left[\theta_n^{(r,k)} - \theta_n^{(r,k+1)} \right] + \sum_{\substack{(r,k),\\ n}} s_{j,n}^{(r,k)} \sum_{j\in\mathcal{N}} \left[\theta_{n,j}^{(r,k)} - \theta_{j,n}^{(r,k)} \right].$$

The last term equals the right-hand-term of the Eq. (13), and therefore completes the proof.