Network for AI and AI for Network: Challenges and Opportunities for Learning-Oriented Networks

Jianping Pan, Lin Cai, Shen Yan, and Xuemin (Sherman) Shen

Abstract

The "data pipe" model used by the existing Internet protocol stack is no longer ideal for many emerging applications, due to multimedia, multicast, mobility, machine learning, and network management challenges. A new learning-oriented network architecture is required to deal with these challenges and serve learning-centric applications in data centers, around network edges, and on mobile devices. This article focuses on the network for AI and AI for network for learning-oriented network architecture. This is done by leveraging, improving, and creating new learning techniques to determine and optimize protocol mechanisms and control policies. The new network architecture can provide ample research opportunities in network topology control, protocol design, and performance evaluation, aiming to network a truly dependable cyber-infrastructure. The learning-oriented network can also learn from applications and communications automatically and continuously while running on different infrastructures to support diverse requirements. In addition, the network can keep evolving its protocol mechanisms and control policies in an online manner. It does this while maintaining protocol security and preserving user privacy, to learn and perform more effectively and efficiently. Finally, the main challenges and opportunities of learning-oriented network are discussed, encouraging further research.

Introduction

Computer networks were initially designed as pipes for data streams, bytes in and bytes out, in order and reliably. The "data pipe" model used by the existing Internet protocol stack is no longer adequate for many emerging applications. Newer applications, for example, multimedia ones, care more about information, not only data representation, so audio and video data can be transcoded if needed inside the network to meet the Quality of Service (QoS) and Quality of Experience (QoE) requirements. Emerging learning-centric applications care more about knowledge extraction and dissemination, so information can be fused inside the network to meet the convergence speed and accuracy requirements. Different from those for point-to-point, reliable, and elastic data transfer, information and knowledge can be aggregated and transformed inside the network, and control scope is no longer limited to endpoint-to-endpoint only.

On the other hand, new communication infrastructures such as 6G mobile communication systems, low-orbit satellites, and underwater/ground communication systems are emerging rapidly, offering additional communication capabilities with certain topological features. Now the network protocol stack between application requirements and communication services becomes a new bottleneck.

Looking back at the history shown in Figs. 1a–1d, ARPANET, funded by the Advanced Research Projects Agency (ARPA) in the 1960s, was the first "domain-specific" packet-switching computer network. The TCP/IP protocol suite, as the basis of today's Internet, was standardized in the 1980s, initially targeting remote login, file transfer, and email applications. In the 2000s, wireless access networks and the Internet were converged to provide Internet services anywhere, anytime. With the Internet architecture and TCP/IP protocol suite, traditional computer networks build data pipes between computers, with strict layering and end-to-end principles, leaving a "smart end vs. dumb network" legacy. They are ill-suited for the new era of integrated Internet for ubiquitous intelligence.

The rise of middle-boxes already breaks the strict layering and end-to-end-only principles in computer networks. The emerging data mining and machine learning applications [1, 2], to transport not only data but also information and knowledge, demand new communication primitives, including multipoint-to-multipoint and in-network processing/aggregation due to control and latency constraints. In this process, networks need to manage not only the bandwidth and buffer, but also computing and caching resources. In addition, these in-network processing and learning changes the traffic flow characteristics and require different types of service requirements. How to make the entire protocol stack smarter and support learning applications, for cloud, satellite, and 6G communication systems is a pressing issue.

Furthermore, today's large-scale ubiquitous network, penetrating deeply into our daily lives, mandates the intelligentization of existing communication infrastructures, that is, being secure, privacy-preserving, smart and dependable [3]. Existing Internet protocols are lacking the ability to self-learn, configure, decompose and compose. They encounter major challenges in supporting new demanding applications with stringent delay, reliability and throughput requirements, especially

Digital Object Identifier:
10.1109/MNET.101.2100118

Jianping Pan are Lin Cai are with University of Victoria; Shen Yan is with Huawei; Xuemin (Sherman) Shen is with the University of Waterloo.

0890-8044/21/$25.00 © 2021 IEEE

for Internet-of-Things applications. Unlike human beings, machine consumers are less flexible and intelligent to handle communication impairments. They often have to rely on ultra-high reliable, real-time information and control instructions to coordinate with each other. In addition to these QoS requirements, distributed AI also brings new security and privacy concerns on how much the applications should expose to the network and vice versa.

In this article, we first review the existing network architecture approaches in the last two decades. The new network design requirements and challenges for supporting AI applications are then presented. Next we discuss the new opportunities applying AI for network intelligentization. Security and privacy issues for supporting distributed AI are then discussed, followed by concluding remarks.

Network Architecture Research Review

The great success of the Internet network architecture and TCP/IP protocol stack, and the challenges faced to support the new multimedia applications and explore new communication capabilities, inspired the research community to explore high-performance network protocols in the 1980/90s and new Internet architectures in the 2000/10s.

Although the Internet is still evolving around TCP/IP, new protocols have emerged above and below IP, being deployed in an overlay or underlay fashion. Networking expands from host connectivity only to data dissemination mainly, with content distribution networks (CDNs) and named-data (NDN) and information-centric networking (ICN). Network/application-layer multicast and host/network mobility have been heavily explored with the advance of wireless communications systems. Security and privacy become the primary concerns when the Internet becomes the critical infrastructure for not only personal and commercial uses but also public safety. Together with multimedia, multicast and mobility, they are the traditional drivers for network innovation; now machine learning also joins the challenge for a more manageable network.

However, the traditional network architecture and protocol stack has not changed much as a whole. Protocol design still follows the layered structure and mostly end-to-end argument, in a one-fits-all manner. The research community and industry had concluded [4, 5] that such a rigid approach will not satisfy today's diverse application requirements and fully explore the existing and upcoming communication systems. But how to break down the protocol stack for more flexible network architecture is still under great debate and needs more research efforts.

Meanwhile, many researchers including ourselves have attempted machine learning approaches to computer networks and have obtained various successes, but still at the patchwork and functionality level, lacking a systematic approach to the architectural mismatch to better support learning-centric applications and leverage learning inside the network architecture itself. In other words, we need a new network architecture design philosophy and protocol stack realization strategy beyond the current Internet and TCP/IP

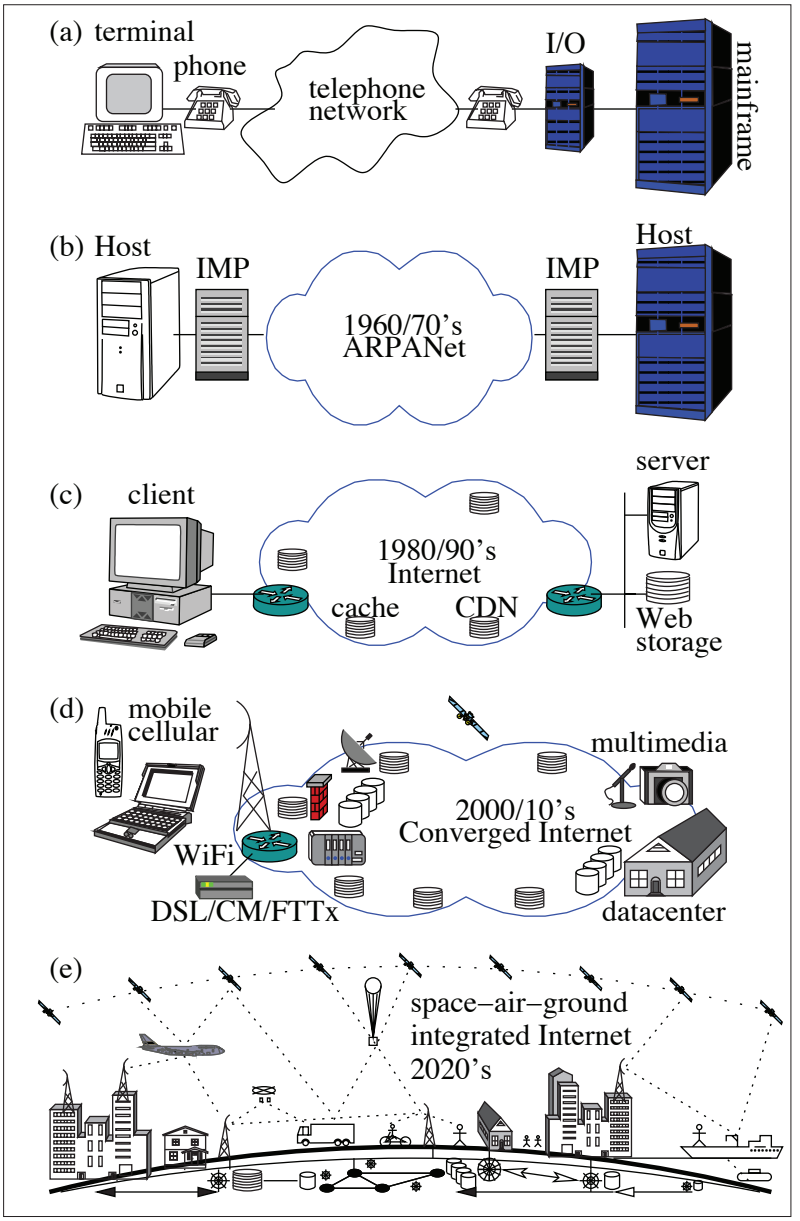

FIGURE 1. Communications and computer network history: a) Circuit switching telephone network; b) Packet-switching ARPANet; c) TCP/IP Internet (Internet 1.0); d) Converged wireless/wired Internet (Internet 2.0); e) Integrated Internet for ubiquitous intelligence (Internet 3.0).

[3, 6], paraphrasing Samuel [7]: *Can we give protocols the ability to learn without being explicitly specified?*

As networked systems become "a giant computer," we can learn from what happened in computer architectures (CA), operating systems (OS), and programming languages (PL) in terms of architectural evolution and revolution. Specifically, *domain-specific*, *modular*, and *object-oriented* CA/OS/PLs such as GPU/TPU/NPU, Linux, and Python have achieved great successes recently, partially fuelling the rapid development of data mining/machine learning, instead of sticking to an all-in-one or one-fits-all approach. For networking, SDN is an existing attempt toward this direction [8].

We have examined the applicability of the domain-specific, modular and object-oriented principle in network architecture, although computer networks have their own features and require-

Not all learning-centric applications can collect and process data in a cloud data center, due to privacy concerns and latency constraints. Thus federated and distributed machine learning has attracted a lot of attention in the last few years. Leveraging the existing trust between users and their immediate Internet service providers (ISP), many learning opportunities can happen at the edge of the network.

ments for interconnection. Essentially, modularity is a powerful approach, not just by layering. If we can decompose and define a set of relatively simple building blocks and compose and assemble them according to application requirements and network dynamics, we can avoid the "networking as a bag of protocols" curse. Here, how to compose and assemble is a new challenge, where learning is the key.

Network for AI Applications: Challenges and Design Requirements

A new learning-oriented network architecture is needed to support learning-centric applications in data centers, around network edges and on mobile user equipment by exploring, improving, and creating effective learning techniques to decompose, compose, optimize and configure protocol mechanisms and control policies for secure and privacy-preserving distributed AI, as shown in Fig. 1e. We present the challenges and design requirements for learning-oriented network architecture from four aspects: high-intensity data centers, large-scale network edges, high-mobility 6G devices, and end-edge-cloud orchestration, which motivate the new learning-oriented network architecture in the next section.

High-Intensity Data Centers

Today many high-intensity data mining/machine learning applications happen in cloud data centers. Traditional connectivity applications mainly incur north-south traffic in points of presence between servers and clients, and current data-intensive applications mainly occur in west-east traffic between servers. However, learning-centric applications with deep neural networks (DNNs) can involve "big data" sources, and "deep" layers of processing and decision nodes in a complex graph topology for frequent forward and backward computation and propagation (e.g., stochastic gradient descent) potentially among many servers in different computer racks, rows and areas in a data center. DNN training needs communication among neurons in different layers with different interconnections in both forward and backward propagation, in an iterative and synchronous or asynchronous manner, which creates new challenges due to one-to-many, many-to-one and many-to-many communication paradigms, so new network protocols need to be designed and optimized to support multicast and incast flows, adaptive precision quantization and update sampling, and so on. Given the complex graph topology involved, the existing optimized solutions for north-south (client-server) or west-east (server-server) traffic are challenged, and the communication between the servers for learning-centric applications can easily become a bottleneck.

The new learning-oriented network architecture should leverage the regularity of data center network (DCN) topologies and communication patterns in learning-centric applications, and incorporate the sampling/compression techniques as not all model parameters and gradients are of the same importance for convergence accuracy and speed. Furthermore, the new architecture should have the flexibility to explore the learning-centric applications between geo-distributed data centers, where the propagation delay and wide-area network (WAN) traffic and cost are major concerns.

Large-Scale Network Edges

Not all learning-centric applications can collect and process data in a cloud data center, due to privacy concerns and latency constraints. Thus federated and distributed machine learning has attracted a lot of attention in the last few years [9]. Leveraging the existing trust between users and their immediate Internet service providers (ISP), many learning opportunities can happen at the edge of the network.

However, each edge network may only have limited views and insights into the learned model due to the limitation in users and data, and the model lacks generality and extensibility. How to make a proper trade-off to distribute and federate learning at the network edge, especially with enough scalability, is still a challenge in the machine learning domain, where networking can also help.

Instead of always transporting the entire data set or model parameters, we can compress/aggregate partial information/knowledge gradually at the edge, in a controllable lossless or lossy manner, to meet the network bandwidth and latency constraints. Our work on edge-based sound event localization and identification shows the promise of this approach [10].

High-Mobility 6G Devices and Networks

6G mobile communication systems will become the Internet of Intelligence (IoI), connecting all kinds of mobile devices above, on, or underground or water. Low-orbit satellites and high-altitude platforms can also become part of the backhaul and backbone networks. Regardless, most of them have mobility, some very high, although a few are very predictable (e.g., satellites). Mobile networks bring new stability and scalability challenges to the IP protocol and IP-based routing, which were designed for fixed network infrastructure. For instance, using a satellite backbone, the high mobility of satellites will lead to frequent changes of IP routing tables, not affordable for satellites with limited energy supply and computing powers.

On the other hand, for many learning applications in 6G, data acquisition and transportation rely on high-mobility end devices (UAVs/AUVs, drones, robots, and so on) and high-mobility access networks (high altitude platform systems, satellites, and so on). How to support these AI applications given both the user and network mobility is a great challenge for network protocol and architecture design. For applications involving mobile users, they may accumulate lots of data around them, but cannot transfer all data to the cloud due to communication and privacy constraints. Edge can help, but for some sensitive

data, users prefer to keep data with them only or in a peer-to-peer or device-to-device manner while still benefiting to and from the learning at the edge and/or in the core. Mobile users and their data will come and go time-wise and show up at different locations, which requires additional network support when end-to-end connectivity cannot be guaranteed in opportunistic sessions. Although mobility brings challenges, highly predictable mobility can be leveraged to improve the data collection performance for learning applications.

End-Edge-Cloud Orchestration

Overall, learning can happen in the data center, around the network edge, and on end-user equipment, and many learning-centric applications may involve all three of these levels. The local, partial knowledge from the device and edge needs to be aggregated for useful and meaningful inference and decision making. This communication paradigm is very different from the traditional point-to-point, reliable, and elastic data transfer, as information can have different data representations, and knowledge can be transformed and transferred. The end-edge-cloud orchestration is required for communication networks, so the network can become a giant, smart computer, with computing, storage, sensing, and control inside.

An example of end-edge-cloud orchestration is given in Fig. 2 for abnormal sound event identification and localization, which consists of audio feature extraction, sound source localization, and sound event classification [10]. Here, the end devices, edge servers, and cloud-based servers can coordinate with each other, so the abnormal event, such as gunfire, can be detected promptly and accurately. End devices (sensor nodes) collect raw audio samples and conduct simple audio signal processing/compression and send mel-spectrogram to edge servers for aggregation and ensemble learning. Meanwhile, the edge servers collect localization information from sensors to weigh learning parameters properly (depending on the distance between the sensors and sound source, their mel-spectrograms have different contributions to sound identification), and avoid over and under-fitting in localization (too much or few ranging information can distort the localization accuracy). Edge servers can relieve the core network congestion while enhancing the privacy of local communities by pre-processing the audio signals for feature extraction and sound source localization, and cloud-based servers have high computation and storage capacity to handle time-consuming post-processing tasks such as sound classification by a pre-trained neural network. Considering the biased audio information due to source-sensor geometries, a cooperative decision-making algorithm will aggregate the sound event classification results with adaptive control and ensemble learning.

The end-edge-cloud orchestration of the above example is coordinated by the edge server to provide low latency services, and it decides to offload computation-heavy and delay-tolerant tasks to the cloud. Generally speaking, in-network elements, such as edge servers, can coordinate data compression and information aggregation, with control signalling interacting with end points and

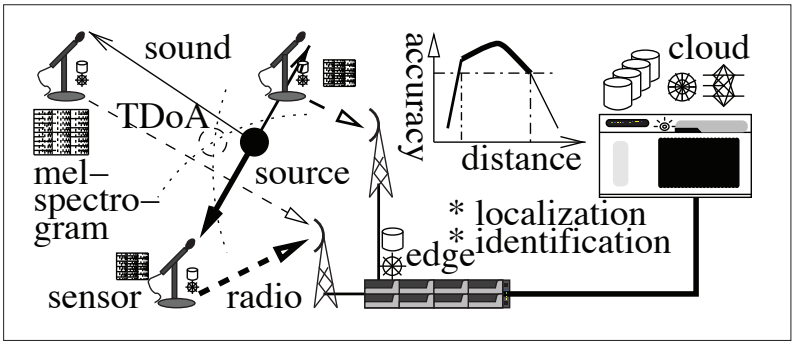

FIGURE 2. Cooperative abnormal sound event detection in the end-edge-cloud orchestrated system.

applications. The new learning-oriented network architecture can incorporate the three-level hierarchy of intelligence, end, edge, and cloud, to support a wide variety of AI applications. Here, efficient end-edge-cloud orchestration at different scopes and with different levels of coupling for distributed AI in computing, communication, and control is a challenging, open issue.

AI for New Network Architecture and Protocols

With the diverse learning-centric applications, the network protocol stack needs to adapt to different application requirements. The traditional approaches, one-fits-all or all-in-one, are no longer viable for both the network and new applications. Decomposing the existing protocol stack, identifying and creating basic building blocks, and only composing and assembling needed ones for certain applications and to adapt to network dynamics, is a more promising approach. This approach can also avoid redundant and sometimes conflicting controls adopted in different layers in the current Internet protocol stack.

How to design and engineer such a flexible architecture and adaptive protocols is a complicated issue, while we can apply advanced AI techniques to assist the process. In the following, we first present a big picture of the new network architecture with AI-powered protocol stack. Several aspects to apply AI for learning-oriented networks are then discussed.

Learning-Oriented Network Architecture

As shown in Fig. 3a, in the early days, terminal-to-controller communications happened in a request interrupt or polling manner to exchange keystroke codes and screen echoes in a character-by-character or block-by-block mode, with asynchronous or synchronous serial or parallel communication lines, possibly through dial-up or leased-line modems. The demand of host-to-host communications driven by remote job entry (rexec), remote login (telnet), file transfer (ftp) and electronic mail applications saw the development of peer-to-peer network protocols such as the ARPANET network control program (NCP) and Internet TCP/IP with symmetric communication endpoints and dedicated interface message processors or routers. With the birth of the World-wide Web in the 1990s, client-server applications dominated the Internet in a request-response

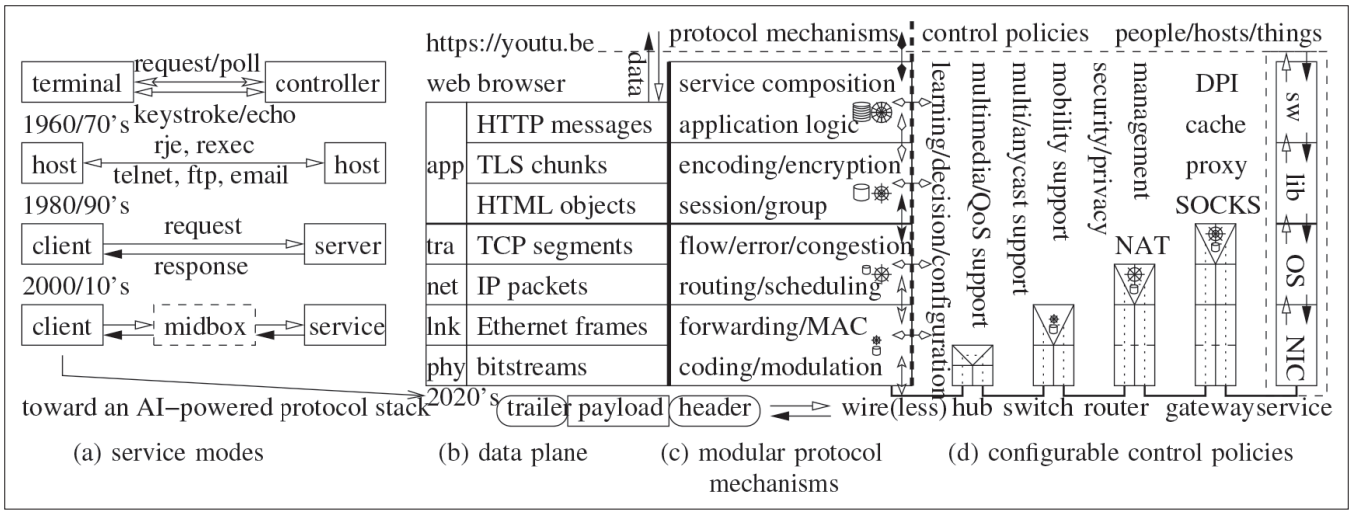

FIGURE 3. Toward learning-oriented network architecture.

transaction manner, and drove the evolution of HTTP/0.9 to 1.0 and 1.1 to run on top of TCP/IP better.

Since 2000, many mid-boxes such as network address translators (NATs), socket proxies (SOCKS), cache servers, deep-packet inspection intrusion detection/prevention systems (IDS/IPS) and firewalls, and so on, started to break the strict layering of the network protocols and the end-to-end arguments. With the rapid growth of network applications and in-network devices, and development of HTTP/2 (SPDY) and HTTP/3 (QUIC), the Internet is evolving toward a more intelligent global network. We start to see the separation of the data plane and control plane, as shown in Figs. 3b–3d.

The learning-oriented architecture further separates the protocol mechanisms and control policies in the control plane, where the protocol modules, regardless if they are implemented in network interface controllers, OS, system libraries or application software, can be adaptively composed, configured and optimized at run time based on application requirements and network dynamics, as shown in Figs. 3c–3d. The new opportunities of applying AI for the learning-oriented networks are in the control plane, that is, how the protocol stack decomposes and composes modular protocol mechanisms, and configures control policies for supporting various applications effectively and efficiently, as discussed below.

Decomposition and Composition for Protocol Functions

The TCP/IP protocol stack with the widely used Berkeley Software Distribution (BSD) Unix socket application programming interface (API) offers two transport-layer services, the TCP-supported stream service and UDP-supported datagram service. TCP has connection management and flow, error, and congestion control all-in-one, while UDP has none of them except transport-layer addressing/multiplexing. Thus, Internet applications can only choose between fully reliable, ordered stream service and unreliable, unordered datagram service, or reimplement connection management and flow/error/congestion control themselves or in software libraries again, for example, the Google QUIC protocol.

Instead, following the domain-specific, modular, and object-oriented design principle for large-scale software systems, we can decompose protocol mechanisms such as connection management and flow/error/congestion control in the OS kernel, and allow applications to configure these building blocks according to certain control policies to fit their needs (Fig. 3).

An example of a decomposed transport layer protocol is given in Fig. 4. Here, the connection management, flow control, error control, and congestion control are decomposed, so the control functions can be selected flexibly. For instance, if an application can tolerate packet losses while preferring timely datagram services, the transport layer protocol can keep the flow/congestion control, while skipping connection management and error recovery (e.g., end-to-end retransmission). If another application requires a high-reliability and low-latency streaming service with smooth throughput, the protocol should carefully configure its connection management, error control and congestion control modules to meet the service requirements. In addition to considering the service requirements, the configuration of the protocol should also consider and react to the network features. For example, if the connection with long round-trip time relies on a highly dynamic and lossy communication link, the end-to-end error control may combine both the forward error correction (FEC) and automatic repeat request (ARQ) mechanisms to provide high-reliable and low-latency services.

Note that the decomposition shown in Fig. 4 is not complete, as new functions can be included for other service requirements, such as delay/jitter control, and different types of protocol mechanisms can be added for each control function, such as rate-based flow and congestion control, timer-based connection management, configurable error control based on traffic classes, and so on.

Furthermore, these protocol mechanisms and control policies have many parameters, for example, the initial congestion window and slow-start threshold in TCP. Existing implementations hard-code default parameters or adjust them after the connection is established and data transfer is ongoing. Once the data transfer is finished and the con-

nection is closed, the new connection for a similar pair of sender and receiver has to go through the same process, which is lengthy for some short-lived flows. A better approach is for the network to learn from experience and set the initial parameters for the new connection in a more informed way. TCP fast open is an attempt toward this direction, but more systematic approaches are needed to handle the transfer of knowledge in both the time and space domains with different granularities, and also for dealing with the multimedia, multicast, and mobility challenges.[1]

Sequential Decision-Making Network Problems

Given the decomposition of protocol functions, a key issue for the new protocol architecture is how to select and configure protocol functions to meet the service requirements and network dynamics. To compose and decompose protocol mechanisms for dynamic and adaptive control policies, and parameter configuration and optimization, there are many decisions to make, some of which are done sequentially and have interdependence between them and over time. These decisions have to explore the unknown and changing environment and exploit the learned knowledge to minimize regret (performance loss due to the lack of prior knowledge).

To solve the sequential decision-making problems, we should consider the trade-off of exploration and exploitation and aim to minimize regret (performance loss) during the exploration and exploitation process. This problem can be formulated as a multi-armed bandit problem, which can be solved with guaranteed performance (with bounded regrets) using various learning algorithms, such as the upper confidence bound algorithm or the Thompson sampling algorithm. For instance, content caching in wireless cellular edge can be formulated as a stochastic combinatorial multi-armed bandit problem with delayed feedback and forced-to-sleep arms [12, 13], so we can solve the problem effectively and efficiently.

In the context of federated and distributed learning in a cloud data center, around the network edge, and on mobile users, we need to make the multi-armed bandit model more generic and powerful in terms of multiple, combinatorial, sleeping/moving arms (i.e., actions), as well as multiple decision-making agents. Overall, this is a promising tool to solve many sequential decision-making network problems, and it is very useful for the composition process of the protocols in the new protocol architecture with modular, configurable protocol functions.

Multi-Agent Online Optimization

In a networked system, multiple players or agents are involved, most likely distributed. How to optimize network performance in an online manner, that is, improving while things are happening, is a grand challenge [14].

Similar to sequential decision-making problems, we can investigate the trade-off of exploration and exploitation using the multi-armed bandit approach. For example, in a highly mobile ad hoc network where no prior knowledge is given and nodes only have limited data packets to send (i.e., without additional routing packets possibly exchanged in advance), Thompson sampling-based opportunistic routing with multiple agents (i.e., routers) can be applied to ensure bounded regrets.

		protocol mechanisms	
function	procedure	handshake	timer ...
connection management	establishment	3–way	set
	maintenance	seq#,ack#	cntdown
	release	2x 2–way	timeout
		window	rate ...
flow control (FC)	negotiation	winsize	(r,b,p)
	update	ACK	(r',b',p')
	enforcement	ack#+win	(r',b',p')
		ARQ	FEC ...
error control (EC)	detection	seq#,cksm	cksum
	notification	ack#,timer	–
	recovery	retransmit	correct
		loss–based	delay ...
congestion control (CC)	detection	seq#	s/rtt/v
	notification	ack#,timer	ACK
	recovery	AIMD FC	AIMD
		TCP+socket	others

FIGURE 4. Decomposed protocol: an example.

Moreover, in a distributed AI system, there are many data sources, and possibly some "weaker" learners due to the limitation of time, space, computation, communication, and storage capabilities. However, if these weak learners can cooperatively coordinate with each other, they can become strong learners overall. In this sense, the network for AI and AI for the network naturally connect the network and distributed AI. However, not all agents are available all the time, due to the distributed manner, and some might be selfish or malicious. Thus, how to learn effectively and efficiently for distributed AI becomes a new challenge, which will also be addressed later.

Auto-Configurable and Self-Evolving Protocols

The learning-oriented network architecture and decomposed and composable protocol stack need a new API for applications, particularly learning-centric applications, so the protocol stack can automatically configure and continuously improve according to applications' requirements and network dynamics. The applications can give some high-level intent at the beginning, and through the interaction between the protocol stack as the agent and the application and network as the environment. Again, multi-armed bandit approaches can play a key role, making the trade-off between exploration and exploitation in an online manner, so the applications do not have to handle all the complexity inside the network and protocol stack. This will make the network more intelligent.

Here, we assume the applications are cooperative and willing to share the intent with the network, so the network can optimize the protocol composition. However, some applications may not be willing to do so. In this case, the protocol stack can infer the application intention by topology and traffic inference as discussed below.

Security and Privacy of Distributed AI

Similar to the security and privacy issues introduced by communication and networking, distributed AI can also bring in new security and privacy concerns, for example, how much information

[1] To this end, the Path Aware Networking Research Group has been formed recently to investigate how to explore path properties by hosts or other entities for selecting between paths or for invoking some of the provided services [11].

The new learning-oriented network architecture and protocol stack can have certain vantage points inside the network and in neighbors if cooperative, to infer network topology and traffic condition with a small amount of probing traffic. With this knowledge, the protocol stack can automatically explore the diversity inside the network.

the applications shall expose to the network and can trust the feedback and instruction from it, and vice versa [15]. Here, we discuss these challenges from the network protocol and architecture design perspectives.

Topology Inference

Networking may go across different domains (autonomous systems, AS), and the neighbor ISPs may appear to be a blackbox. Certain knowledge about the topology of neighbor networks can help the current network better utilize its resources and serve its users and applications better. Thus the new learning-oriented network architecture and protocol stack can have certain vantage points inside the network and in neighbors if cooperative, to infer network topology and traffic condition with a small amount of probing traffic. With this knowledge, the protocol stack can automatically explore the diversity inside the network, for example, multi-homing, multi-link, multi-hop, and multi-path in the network, without revealing all network details.

Traffic Inference

Not all applications can and are willing to give traffic specifications when creating the flow and during data transfer. Existing network management relies on packet inspection, but end-to-end encryption makes it infeasible. However, the network can infer traffic behavior from packet size, sending gap, flow duration and volume, for example, distinguishing very short liveness heartbeats, medium-sized regular voice packets, and longer video packets. The network can also correlate the incoming and outgoing traffic in timing and length to infer client, server, or relay nodes. Even if the application is not revealing its traffic specification, learning-based traffic inference can help the network to manage it better.

Protocol Security

A smarter protocol stack may incur more security vulnerabilities, and here we focus more on the protocol itself, instead of the implementation due to buffer overflow and other issues. The protocol shall be correct without livelock or deadlock and not susceptible to denial of service (DoS) attacks. Certain model checking tools can be used, while we can also apply learning-based approaches to improve protocol security by watchdog modules, so possible vulnerabilities and attacks can be mitigated before they are exploited by malicious users.

User Privacy

With learning, more application and user information might be exposed to the network and possibly curious or malicious third-parties, and thus user privacy becomes the main concern. Without properly preserving user privacy, users are unlikely to contribute to and thus benefit from the learning, degrading a win-win situation to a lose-lose one.

In addition, after deploying AI applications on networks, AI models should be released and accessed by authorized entities or publicly. Even though only black-box access is allowed, that is, the AI model training algorithm is not disclosed publicly, adversaries can launch many malicious attacks, including membership inference attacks and model extraction attacks, to break the model privacy requirements. Consequently, different network data centers may not be willing to publish their well-trained AI models, which are often considered as intellectual property. Thus the new network architecture and protocol stack have to preserve user privacy including data and model privacy while maintaining protocol security.

There have been extensive research efforts in data and model privacy preserving using cryptography tools like secure multi-party computations and homomorphic encryption. These cryptographic tools can help but still suffer from high complexity and overhead.

We can address the problem from additional approaches such as masking and consensus to achieve differential privacy. However, the masking and consensus process has to tolerate users coming and going, as well as curious and malicious users.

Attacks on AI Security Functions

AI network security functions, for example, AI-based firewall and intrusion detection, can monitor and inspect network traffic intelligently to identify anomalous or malicious activities by learning from historical data. However, AI-based network security functions are vulnerable to malicious adversarial attacks, which generate adversarial inputs to mislead the training of AI models. How to detect and react to these attacks on AI models used to protect the network remains an open problem. Traffic inference mentioned in the previous subsection and generative adversarial models may be useful for countermeasure.

Conclusions and Further Discussions

Network architecture research is challenging and there are many "failed" attempts already [5], but we can learn from the research process and still apply many techniques in new environments, for example, Asynchronous Transfer Mode traffic management now in Multiprotocol Label Switching traffic engineering. More importantly, we have much more confidence in the learning-oriented network architecture for the following reasons.

First, unlike many previous attempts that tried to replace the Internet as a whole, we can focus on specific networks such as DCN, satellite, and 6G networks, where they are relatively standalone by themselves and isolated from others. They can interconnect with the current Internet through a protocol gateway or overlay. Second, although previous attempts tried to address the challenges faced by the Internet from different aspects, they followed the traditional design philosophy for one-shot optimization in a one-fits-all approach. Here, we focus on a learning-oriented approach with modular decomposition and composition. Third, we mainly target learning-centric applications, and we find they will become more dominant in the next decade. Unlike another data pipe, we introduce multipoint-to-multipoint in-network processing/aggregation for learning inside the network

to support learning-centric applications and other applications better.

The learning-oriented approach is expected to have a significant impact to support AI applications in the data center, around the network edge, and on mobile devices, as we expect more of these applications in the coming decades. Different from the previous work that only explored learning above the network, we advocate learning inside the network, to make the network smarter and more dependable, for not only learning-centric applications but also other applications with improved performance and functionality. The learning-oriented network architecture and reconfigurable, self-evolving protocol stack will bring a new golden age for integrated computing and communication networks, similar to that for computer architecture. Extensive research is beckoned to create new learning algorithms, methods, and prototypes, particularly for data center, satellite, and 6G networks. They will have substantial impacts on standards and product development for the future paradigm of Internet-of-Intelligence.

Different from the previous work that only explored learning above the network, we advocate learning inside the network, to make the network smarter and more dependable, for not only learning-centric applications but also other applications with improved performance and functionality.

References

[1] H. Barua and K. Mondal, "A Comprehensive Survey on Cloud Data Mining (CDM) Frameworks and Algorithms," *ACM Computing Surveys*, vol. 52, no. 5, 2019, pp. 1–62.

[2] J. Verbraeken *et al.*, "A Survey on Distributed Machine Learning," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, 2020, pp. 1–33.

[3] Y. Cheng *et al.*, "Bridging Machine Learning and Computer Network Research: A Survey," *CCF Trans. Networking*, no. 1 2019, pp. 1–15.

[4] J. Day, *Patterns in Network Architecture: A Return to Fundamentals*, Prentice Hall, 2007.

[5] D. Clark, *Designing an Internet*, MIT Press, 2018.

[6] R. Boutaba *et al.*, "A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities," *J. Internet Services and Applications*, vol. 9, no. 1, 2018, pp. 16.

[7] A. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM J. Research and Development*, vol. 3, no. 3, 1959 pp. 210–29.

[8] S. Shenker *et al.*, "The Future of Networking, and the Past of Protocols," Oct. 2011, invited talk at Open Networking Summit.

[9] W. Lim *et al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Commun. Surveys & Tutorials*," vol. 22, no. 3, 2020, pp. 2031–63.

[10] J. Wang *et al.*, "Cooperative Abnormal Sound Event Detection in End-Edge-Cloud Orchestrated Systems," *CCF Trans. Netw.*, no. 3, 2020, pp. 158–70.

[11] Path Aware Networking RG (panrg), available https://datatracker.ietf.org/rg/panrg/.

[12] Z. Huang *et al.*, "Caching by User Preference with Delayed Feedback for Heterogeneous Cellular Networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, Mar. 2021, pp. 1655–67.

[13] L. Zhao *et al.*, "Adaptive Content Placement in Edge Networks Based on Hybrid User Preference Learning," *Proc. 2019 IEEE GLOBECOM*, Waikoloa, HI, USA, 2019.

[14] L. Lei *et al.*, "Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges," *IEEE Commun. Surveys and Tutorials*, vol. 22, no. 3, 2020, pp. 1722–60.

[15] X. Shen *et al.*, "Data Management for Future Wireless Networks: Architecture, Privacy Preservation, and Regulation," *IEEE Network Mag.*, vol. 35, no. 1, 2021, pp. 8–15.

Biographies

Jianping Pan [S'96, M'98, SM'08] is a professor of computer science at the University of Victoria. His research interests include protocols for advanced networking, performance analysis of networked systems, and applied network security. He received the Telecommunications Advancement Foundation's Telesys Award 2010, the JSPS Invitation Fellowship 2012, and the NSERC DAS Award 2016.

Lin Cai [S'00, M'06, SM'10, F'20] is a professor with the Department of E&CE at the University of Victoria. She is an NSERC E.W.R. Steacie Memorial Fellow, an IEEE Fellow, and a College Member of the Royal Society of Canada. Her research focuses on network protocol and architecture design supporting multimedia traffic and IoT.

Shen Yan received the Ph.D. degree in computer science from Beijing University of Posts and Telecommunications in 2014. He is the Principal Engineer of the Network Technology Laboratory, 2012 Lab, Huawei Technologies Co., Ltd. and in charge of standardization and promotion works.

Xuemin (Sherman) Shen [M'97, SM'02, F'09] is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research focuses on network resource management, wireless network security, AI for networks, 5G and beyond, and vehicular ad hoc networks. He is a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Chinese Academy of Engineering Foreign Fellow. He received the R.A. Fessenden Award in 2019 from IEEE, Canada; the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society; and the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society.