Rethinking IP Core Networks

Saurav Das, Guru Parulkar, and Nick McKeown

***Abstract*—The Internet core today is completely based on IP routers. Circuits are only used to provide static point-to-point optical links between routers. As others have recognized, current practice makes it hard to take advantage of very high-speed, low-power optical circuit switches in the core. Despite many proposals to mix packet switching with dynamic circuit switching, none have taken hold. In these times of rapidly increasing traffic, and rapidly decreasing profits for ISPs, it is time to rethink how backbones are built. In this paper, we propose to replace the largest backbone routers with much cheaper hybrid packet-optical switches in a fully meshed IP core. We take advantage of new control planes for software-defined networks to control the packet and circuit networks from a single vantage point. To demonstrate the enormous potential cost savings, we present a detailed analysis of the capital expenditure and show how our approach offers 60% cost savings for typical backbone operators.**

***Index Terms*—Computer networks; Internetworking; Optical communication equipment; Wide-area networks.**

I. Introduction

The infrastructure of the Internet core has remained largely unchanged since its invention. It has simply grown larger—more/bigger routers, more/faster links, and greater switching capacities—but still consists of IP routers connected by static optical links. Indeed it is a tribute to the industry as a whole that such scale has been sustained for over two decades.

Sustaining the core comes at a heavy price. A *single* state-of-the-art fully loaded backbone router (BR) today consumes more than 10 kW (more for cooling) [1], and costs more than $1 million. Long-haul backbone links are overprovisioned (4× is typical) to prepare for future traffic growth and unexpected failures. In the face of growing traffic, service providers must keep investing in bigger and faster routers and links, even though revenues are growing quite slowly. As stated recently by a senior Verizon executive: "unacceptably high cost escalations (in the present mode of operation), result in a nonsustainable business case" [2].

There are several reasons for the high cost of current backbone networks. In this paper we will show—through an analysis of the capital costs (Capex) of today's backbone—that 46% of the cost of the entire network is spent on router ports connected to BRs. These ports cost three times as much and number nearly 50% more than access ports. Using *cheaper* or *fewer* BR ports would significantly reduce total network cost. There are three main reasons why BRs have so many ports:

1) *High volumes of transit traffic*: 55%–85% of the packets processed by a BR are just passing through to another BR ([3,4], Subsection II.B]; the packets do not originate or terminate at this location. There is no need to process these packets if the entire flow can be switched directly to the correct BR.
2) *Dimensioning for Recovery*: We can reduce the amount of transit traffic by connecting the BRs in a full or partial mesh. Our analysis shows that in the limit, this approach reduces Capex by about 10%–15% (Subsection II.B); we still need transit links to be in place in case the direct connection fails.
3) *Overprovisioning*: Links in IP networks today are overprovisioned to account for uncertainties in traffic, such as sudden traffic surges or the emergence of new bandwidth-heavy applications. Overprovisioning naturally increases the number of BR ports.

The problem is well understood by the optical/transport networking community, and they have proposed several methods in the past decade to reduce the number of BR ports [3–6]. Proposals include 1) keeping transit traffic in the circuit domain (via optical bypass) instead of letting it touch core-router ports, 2) performing recovery in the optical layer, and finally 3) provisioning optical circuits on demand to reduce overprovisioning.

While optical bypass has indeed been deployed, it has been done in a very static way—once a bypass has been added, it is never changed. In Subsection II.B, we show that this approach results in only 10%–15% Capex savings. Recovery in optical networks to support IP networks is no longer used—all recovery is performed in IP, and IP networks have never used circuits dynamically on demand. And so, core networks continue to be packet switched and the use of circuit switching is limited to provisioning *static* point-to-point WDM links. Prior proposals assume a hierarchy of switching layers—packets running inside circuits—which means repeating functionality across layers (e.g., both layers must implement routing and failure recovery), and leads to bad interactions between them (e.g., packet-routing protocols that become unstable in a dynamic-circuit-switched topology or simply do not make sense when routers are connected in a full mesh).

In this paper, we propose a new architecture for IP cores that involves three key elements that have never been

Manuscript received May 28, 2013; revised September 20, 2013; accepted October 4, 2013; published 00 MONTH 0000 (Doc. ID 191340).

The authors are with Stanford University, Stanford, California 94305, USA (e-mail: sauravdas@alumni.stanford.edu).

http://dx.doi.org/10.1364/JOCN.99.099999

proposed before: 1) in the data plane, we replace backbone routers with hybrid packet-optical switches that have both packet-switching and circuit-switching capabilities in nearly equal measure; 2) we create a full mesh of router adjacencies, where every access router (AR) is one hop away from every other AR; and 3) in the control plane, building on the ideas from software-defined networking (SDN) [7], we create a single converged control plane for both packets and circuits. Together, as we will see, our architecture significantly reduces the number of backbone router ports, reduces the network-wide Capex, and overcomes the problems with previous proposals.

To support the practicality of the idea, we prototyped our approach, and to quantifiably measure its benefits, we performed a Capex analysis and compared its cost with the traditional IP-over-WDM design. With our approach, the fraction of the total cost attributed to backbone ports drops from 46% to 3%, and the overall Capex is reduced by 60%. Furthermore, our approach is almost insensitive to the traffic matrix (TM), making the backbone less vulnerable to new applications. We find the cost scales at a slower rate ($11m per Tb/s versus $26m per Tb/s) if the overall traffic grows to five times the original number.

A. Paper Organization

In Section II we determine the cost of an IP-over-WDM network. In Section III, we describe our architecture for a packet and dynamic-circuit core—how it functions and how it addresses concerns with previous proposals. We describe a prototype implementation in Section IV. We model and analyze the cost of our proposed design in Section V and compare it to the design from Section II. Finally, we present related work in Section VI and conclude in Section VII.

II. IP-Over-WDM Model

So that we can make an apples-with-apples cost comparison between different backbone designs, we use each approach to design AT&T's US IP core network (as reported by Rocketfuel [8]). It is worth noting that, while our design methodology is detailed and comprehensive in its steps, it does not involve any optimization to, say, route packets over diverse paths, minimize the number of optical transponders in the transport network, or use MPLS traffic engineering (TE). We do not pursue optimization because optimization is not the goal here. Instead we wish to obtain ballpark-numbers for the relative comparison of the two architectures. The newest networks also use 40G, 100G, and both the C and L bands. Here, to compare apples with apples, we assumed 40 waves at 10G in both architectures.

Figure 1(a) shows the placement of 16 PoPs across the U.S., aggregating the traffic from 89 other cities. Each PoP consists of multiple ARs dual homed to two BRs. The PoPs are connected by 34 long-haul *edges*, where each edge consists of multiple 10 Gb/s links. The ARs may or may not be located in the same city as the BR.

The BRs are connected by a fiber/WDM network, for which we assume the topology from [4] shown in Fig. 1(b).

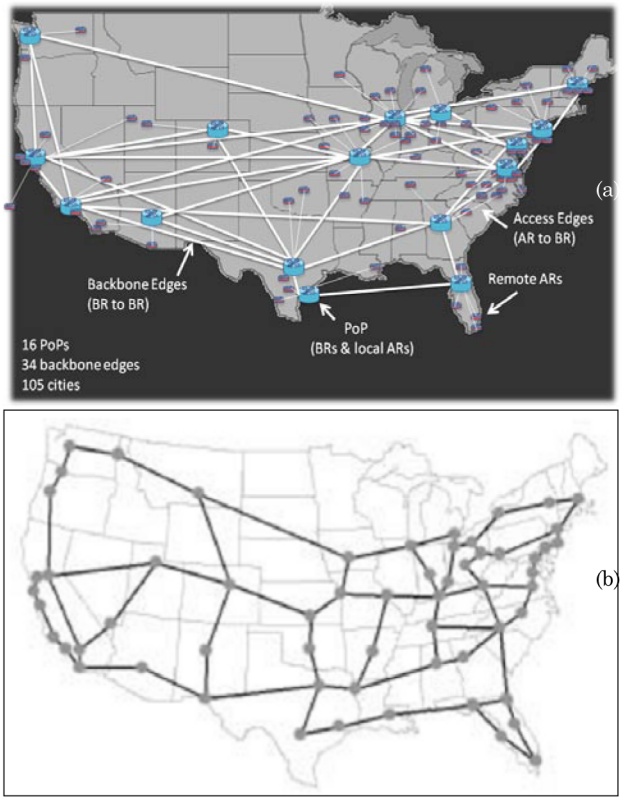

Fig. 1. (a) IP topology. (b) Fiber/WDM topology.

The fiber topology has 60 nodes and 77 edges. The edges are constructed from multiple parallel fibers and wavelengths, with a maximum of 40 wavelengths per fiber.

Note that the graphs are quite different. The IP link is a logical link running over an optical wavelength path (circuit) stitched together from two or more point-to-point WDM line systems. The mapping between an IP link and the "wave" is static, as there is no active *switching* in the underlying optical transport network.

A. Design Methodology

In order to determine the cost of the IP-over-WDM network, we need to complete the design. We follow the approach in [5], which has the following steps.

1) Unidirectional AR → AR Traffic Matrix: First, assume a gravity-model TM [9]. In Section V, we will vary the TM and study its effects on Capex. For each of the 105 cities in the IP topology, we estimate the traffic sent to the other 104 cities. We then scale the TM entries to achieve a cumulative traffic demand on the IP network of 2 Tb/s. (From discussions with ISPs, 2 Tb/s appears to be a reasonable estimate of current aggregate traffic demand on a large U.S. backbone network.)

2) IP Edge Dimensioning: Next we need to determine how much IP traffic will traverse an edge in the IP topology so we can decide how big the WDM circuit needs to be. The IP traffic includes 1) the average traffic demand between cities routed over the edge, 2) traffic rerouted over the edge

in the event of failures, and 3) head room (overprovisioning) for variability in the traffic volume. To pick the speed of the link we do the following:

1) Assume all IP traffic between ARs is routed using Dijkstra's SPF algorithm. From this, we determine how much traffic flows between each pair of BRs.
2) Next, we decide how much extra capacity to provision so we can recover from failures. We consider the capacity we need if we break each edge and node in the IP backbone topology (one at a time) and reroute the *entire* TM over the remaining topology. This emulates how the routers would reroute traffic too, and so gives us an estimate of the amount of traffic each edge must carry. When added to the result in 1), it tells us how big each link should be.
3) Finally, we overprovision the edges to prepare for traffic variability and growth. We chose a utilization factor of 25%, which translates to 4× overprovisioning.
4) AR-to-BR edges are dual homed for recovery and are similarly overprovisioned.

3) IP PoP Dimensioning: Next we determine how many routers (ARs and BRs) we need in each PoP and the number of links that make up an edge. The number of parallel access and backbone links (or ports) can be determined from the edge demand by accounting for the line rate of a single interface (assumed 10 Gb/s). The number of core routers in each PoP is determined by summing up all the access and core interface capacities, and dividing by the switching capacity of a BR (assumed 1.28 Tb/s). Similar calculations apply to the ARs.

4) WDM Network Dimensioning: Finally, we determine how the IP links are mapped to WDM waves, and from that determine how many waves we need on each edge. First, we assume the IP traffic follows the shortest path (in miles) over the WDM network. We then assume each wave carries 10 Gb/s of traffic, and deduce how many waves we need, and therefore the total number of WDM line systems. We also account for the "optical reach" of the WDM line system (assumed 750 km), fully and partially lit systems, WDM transponders with client and line-side transceivers, and optical components, such as amplifiers, wavelength mux/demux devices, and others. More details of all the steps in the design methodology and the assumptions made in each step are discussed in [10].

Table I shows our design for the AT&T IP network. The 48 BRs collectively have 2564 core-facing interfaces and 1776 access-facing interfaces that connect to 232 ARs. From our discussions with ISPs, core networks with a few hundred routers are typical. If anything, we underestimate the number of router and port counts, due to our somewhat simplified PoP structure.

TABLE I
IP Network Dimensioning Results

City-PoP	BRs	Core_ Intfs	Local ARs	Local AR Intfs	Remote ARs	Remote AR Intfs
Seattle	2	48	1	4	4	6
San Francisco	2	120	1	18	16	110
Los Angeles	6	288	6	232	11	76
Phoenix	2	68	2	64	5	24
Denver	2	20	1	6	4	4
Dallas	4	276	2	48	6	20
Houston	2	120	2	44	7	36
Orlando	2	120	1	16	18	120
Atlanta	4	236	1	12	12	28
St. Louis	4	256	1	12	16	28
Chicago	4	392	2	74	22	40
Detroit	2	60	1	42	9	30
Washington, DC	2	152	1	10	19	50
Philadelphia	2	88	1	28	10	24
New York	6	248	7	298	32	192
Cambridge	2	72	4	32	10	48
	48	2564	31	940	201	836

B. Capex Analysis

To determine the overall Capex, we need to know the cost of each component—information that is usually covered by confidentiality agreements and is not made publicly available. However, we believe [6] is a good reference for a detailed and comprehensive price model of IP and WDM systems. It includes the cost of router chassis, slot and port cards, and WDM equipment. The type of parts chosen in our analysis, their exact usage, and their relative cost are described in detail in [10].

By applying the price model to the numbers shown in Table I we calculate the Capex cost for our IP-over-WDM design (Fig. 2). The WDM network routes 1268 waves at a cost of $18.138 million; 77% of the WDM network cost is attributed to the WDM transponders, and the rest to all other optical components. Since we assume we use the same interface (10GE) on the ARs and BRs when connecting them to each other, the cost is the same for each. However, the core-facing ports on the BR are much more expensive and make up nearly half (46%) of the cost of the entire network ($34 million out of $74 million)!

The reason the network is so expensive is because of the large overprovisioning needed to prepare for failed links.

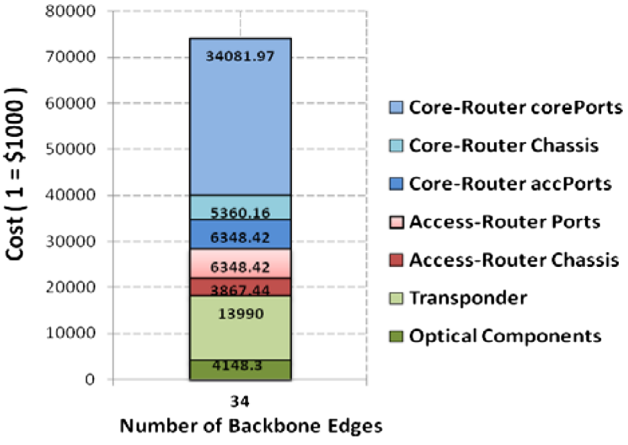

Fig. 2. Capex results.

For example, the Orlando PoP sees transit traffic grow from 0.92 Gb/s under normal operating conditions to 136.18 Gb/s if a link fails. A large percentage of the switching capacity for each BR is sitting unused, waiting for a link to fail. This is very inefficient. In more than half of the PoPs, the unused capacity is 55%–85% larger than needed; see Fig. 3.

It is easy to see why optical switch vendors have proposed optical bypass (sometimes called express links) to reduce the amount of transit traffic processed by the BR. In optical bypass, transit traffic remains in the optical layer (WDM) when passing through a city PoP and does not reach the IP layer. The bypass can be manually implemented with a patch chord or with a more expensive wavelength switch. Typically the bypass is static and is set one time and never changed.

To model the benefits of adding bypass switches, we increase the number of edges in the IP topology [Fig. 1(a)] from 34 up to a maximum of 120 edges. Figure 4 shows how the Capex varies with the number of extra edges. We find that as we increase the number of edges, there is a reduction in Capex, but after we remove large transit traffic, we run into diminishing returns. The overall cost does not reduce much after we have 50 edges, including going all the way to a fully meshed IP PoP topology of 120 edges (Fig. 4).

The main reason is that while the aggregate-transit traffic bandwidth can be reduced, it cannot be eliminated. For example, consider the case when the BRs are fully meshed. When there are no link failures, all of the BRs are one hop apart. But after a failure, traffic has to be redirected over a two-hop minimum, creating transit traffic. Later we will also see that the bypass decisions made for a particular TM do not work well for other TMs. Since the underlying optical network is static, it cannot change with traffic needs. Thus the IP network has to *overprovision in advance* for such change, thereby further increasing costs.

The main takeaway is that static optical bypass can incrementally reduce Capex by 10%–15% and is limited by our need to reroute around failures and prepare for changing TMs. In the next section we describe a core network based on *dynamic* circuit switching (DCS) instead.

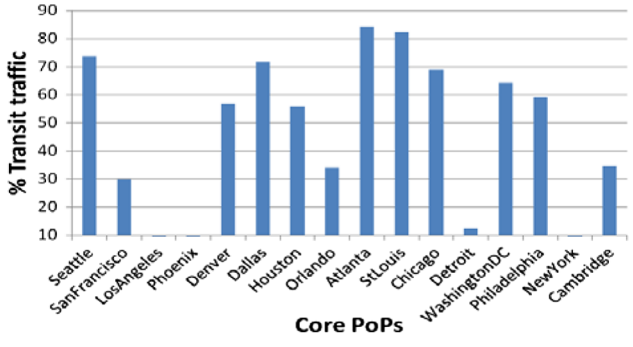

Fig. 3. Transit traffic.

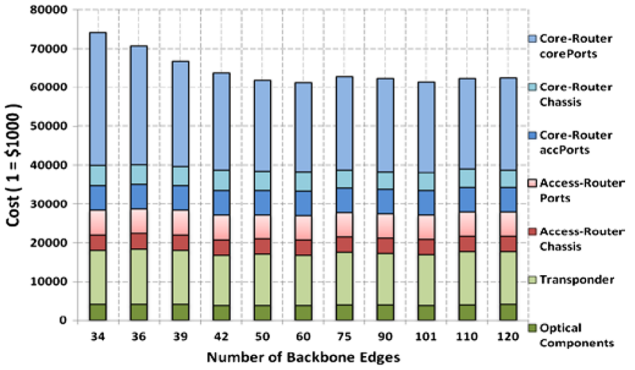

Fig. 4. Effect of adding bypass links in IP topology.

III. Packet and Circuit Core

We now describe a very different design for a backbone network, based on both packet and DCS. We call the design "IP-and-DCS" to distinguish it from IP-over-WDM. We introduce our architecture, describe how it works, and describe how it overcomes practical problems that have held back previous combined packet and optical switching designs. In the next section we determine how much it costs, so we can compare it with our IP-over-WDM design.

A. Architecture

Our approach has three main parts:

- Replace BRs with hybrid packet-optical switches.
- Connect the PoPs in a full mesh.
- Use an SDN-based control plane for both packet and optical switching.

1) Packet-Optical Switches: The basic idea is to keep all transit traffic in the circuit domain during normal operation *as well as during failures*. We propose not just a reduction, but the complete *elimination* of all core-facing BR ports, and we replace the BRs with hybrid switches that have both packet-switching and optical-switching fabrics in nearly equal measure (Fig. 5). All of the packet switching in the hybrid switch happens on the interfaces to the AR, and the ARs continue to be dual homed to the hybrid switch. All of the core-facing ports are optically circuit switched.

The hybrid switch has 1.28 Tb/s of total switching capacity, half of which is an MPLS packet switch, and the other half an OTN circuit switch [11]. The MPLS part switches traffic between the ARs in the same PoP. The MPLS part aggregates and deaggregates traffic to and from other PoPs. Aggregated traffic is forwarded to the OTN part via virtual ports (Fig. 6). Between the two switching fabrics, hardware maps packets to and from time slots. The OTN is a time-slot cross-connect, mapping time slots from the AR to available time slots on the core-facing ports, and vice-versa.

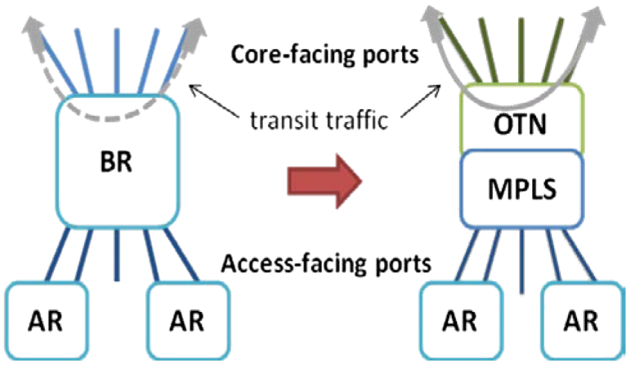

Fig. 5. Replacing BRs in core PoPs with hybrid MPLS-OTN (packet-optical) switches.

Transit traffic from other PoPs is *not* switched by the MPLS part of the hybrid switch; it merely gets mapped from one core-facing time slot to another in the OTN. Keeping transit traffic in the circuit domain means less packet switching, and substituting it with cheaper, lower-power circuit-switching fabrics and interfaces. We can keep transit traffic in the circuit domain even under failure because the circuit switches are reconfigured dynamically.

2) Full-Mesh Topology: A direct consequence of keeping all transit traffic in the optical circuit domain means that the PoPs are fully meshed. Additionally, the circuits are dynamic; i.e., it is possible to alter (by redirecting) the amount of bandwidth between a pair of PoPs at any time.

There are several advantages to dynamic full-mesh connectivity. It trivializes routing and greatly simplifies recovery. Because of the full mesh, all the ARs are essentially one hop away from each other in the *entire* network (i.e., the ARs form a fully meshed IP topology). While the connectivity between PoPs may be built from multiple individual circuits, from the point of view of the IP network (i.e., the ARs), this quantization is not visible. All the ARs see a single-hop path to every other AR.

3) SDN-Based Control: Finally, our proposed design uses an SDN control plane [7]. Briefly, SDN advocates the separation of data and control planes in networks. The data plane is controlled by a well-defined vendor agnostic application programming interface (API), such as OpenFlow [12]. The data plane is controlled by a remote controller or network operating system (NetOS). The NetOS maintains an up-to-date view of the network state, which we can think of as an annotated graph of the topology. All network control functions (such as routing, TE, recovery, etc.) run as applications on top of the NetOS, and manipulate the controller view. The NetOS translates the map manipulations into data-plane reality by *programming* the data-plane switch *flow* tables via a switch-API such as OpenFlow.

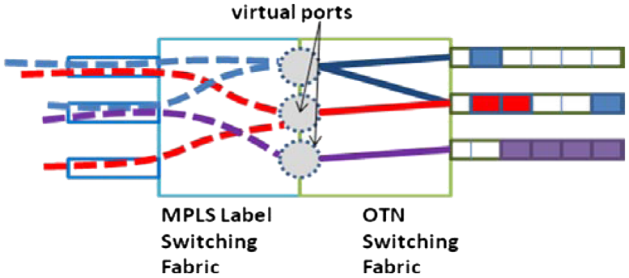

Fig. 6. Packet-optical switch internals.

In our design, all switches including the ARs and the backbone hybrid switches support a generic packet switch and circuit switch *flow abstraction* manipulated by a common switch-API. Further, a (distributed) controller creates a *common-map abstraction* so that network control functions can be implemented and jointly optimized across packets and circuits from a single centralized viewpoint (Fig. 7). Additionally, by pulling the decision making out of the routers, SDN obviates the need for distributed routing protocols, as routers no longer make routing decisions. A WAN would still require multiple physical geographically distributed controllers, but they need not use distributed routing protocols (such as OSPF and IS-IS) for sharing state. Instead they can make use of techniques and advancements made in the distributed systems community for sharing state amongst distributed servers [13]. We discuss the need for SDN in more detail in Subsection III.C.

B. Functional Description

Let us see how the SDN control plane implements routing, recovery, and congestion avoidance.

1) Routing: In our design, the routers and switches learn about advertized IP prefixes from the controller. The controllers maintain E-BGP sessions with neighboring ASes. They discover the intra-AS topology (using the switch-API) and figure out the BGP-next-hop within the AS for all IP-destination prefixes. The route calculation to the

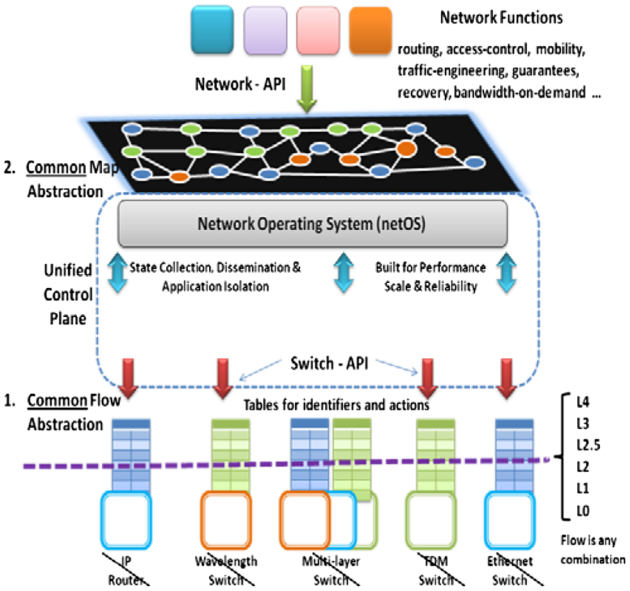

Fig. 7. SDN-based unified control architecture [10].

BGP-next-hop is trivial, as all ARs are one hop away from all other ARs in the full-mesh IP topology.

Each AR pair is assigned a set of globally unique MPLS labels. Traffic between AR pairs can always be identified with the associated labels. Once the controller decides to route an IP-destination prefix to a destination AR, it writes a rule into the source AR's flow table. The "match + action" rule matches on the IP-destination prefix and performs the following actions: 1) it pushes the MPLS label corresponding to the source and destination AR pair, and 2) it forwards the packet to any directly attached hybrid switch in the same PoP.

The hybrid switch maintains one or more circuits to every other PoP. The controller writes a rule into the hybrid switch to match on incoming MPLS labels and forwards packets to the correct circuit to reach the destination PoP. The flow of a packet from an AR in PoP A to an AR in PoP B is then the following: the packet is given a label at the first AR and sent to the first hybrid switch. The packet is placed in the circuit for the destination PoP, where it is removed from the circuit and sent to the destination AR.

The label-to-virtual-port bindings do not change frequently; behind the scenes, if there is a change in network topology (discussed next), it is the circuits and not the virtual ports that change. In other words the circuits may be resized or rerouted, but they remain pinned to the same set of virtual ports. Further, as new IP prefixes are learned (via E-BGP sessions), the controller downloads the prefix/destination-AR/label information into all ARs without having to perform a shortest-path calculation or changing the rules in the hybrid switches. Note that rules are written into the switches and routers proactively: there is no need to use the reactive method of OpenFlow control—if incoming packets do not match an existing rule, they are dropped by the router or switch.

SDN-based control greatly simplifies the backbone switches in hardware and software. The hybrid switch needs a small label forwarding table in hardware similar to today's backbone IP routers that use MPLS forwarding and BGP-free cores. The FIB needs to be on the order of a few labels per egress AR (hundreds). More importantly, in our design there is no need to support distributed routing, signaling, and label-distribution protocols supported in all backbone routers today. This results in simple and inexpensive switches, which can be modeled by today's carrier Ethernet switches (from [8], as reflected in our Capex analysis in Section V).

2) Recovery: We use a two-step recovery scheme in our design. As all core-facing ports are circuit ports, all core network recovery can be handled in the dynamic-circuit layer. Recovery in MPLS-TE or optical networks is always preplanned where primary and secondary backup paths are precalculated and results are cached in the switches. We use a similar scheme in which the results of well-known recovery techniques such as shared-mesh restoration [14] are preprogrammed into the switches. Essentially shared-mesh restoration involves the use of spare capacity in the form of predetermined backup paths, which are shared by multiple disjoint primary paths. In other words, capacity from the "mesh" is used (shared) to redirect traffic around the failures. When failures happen, the switches use the preprogrammed rules to failover to backup paths. This is fast and does not depend on communicating with the controller.

However, the preprogrammed backup paths are notoriously hard to optimize as one has to plan for every single failure scenario (NP-complete). But the advantage of the preprogrammed backup paths is that they buy the controller time to optimize the recovery paths for flows. Thus as a second step, the controller optimizes rerouting of flows with full view of the network map, and full knowledge of the network state, as now the exact nature of the failure is known. Thus recovery happens in two stages: fast preprogrammed failover followed by slower but optimized rerouting; and all with cheaper circuit resources.

3) Overprovisioning: All networks perform poorly when congested, and some overprovisioning is necessary. In our design the overprovisioning of the core network is all in the circuit domain. In Fig. 8(a), the routers have four core-facing interfaces each, presumably where only two are required to satisfy the demand traffic, while the other two are a result of overprovisioning. In our design [Fig. 8(b)], the hybrid switches still have four core-facing interfaces each, but the interfaces are cheaper circuit ports. The quantization of circuits is not visible to packet traffic. Paths to other PoPs appear as big circuits, within which packet traffic can grow and diminish seamlessly. Thus the total bandwidth is the same as the IP-over-WDM network, but is cheaper to deploy because it does not require packet-switched ports.

Additionally we continue to take advantage of dynamic circuits to redirect bandwidth between the core-facing circuit ports from other parts of the mesh network [Fig. 8(b)].

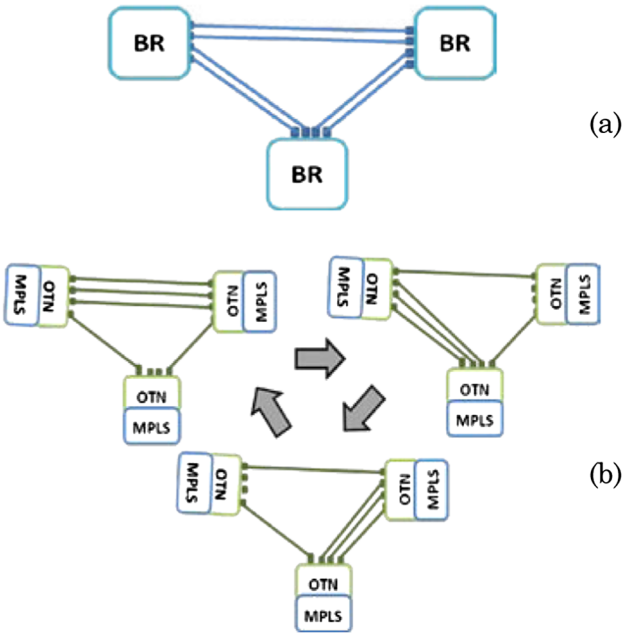

Fig. 8. (a) Fixed IP topology. (b) Bandwidth-on-demand.

Dynamic circuits allow redirection of bandwidth where and when needed to deal with traffic uncertainties. At different times, due to a variety of reasons, bandwidth can be increased between any PoP pair. Such reasons could include congestion, recovery, time-of-day needs, or service/application needs. In short, overprovisioning and dynamicity make the design insensitive to varying traffic or service patterns, while still achieving lower costs.

C. Advantages Over Previous Attempts

We now discuss why our approach overcomes the limitations of previous suggestions for combined packet- and circuit-switched core networks.

1) Solving the Redundancy Issue: The decision to handle all recovery in the circuit layer brings up an important point. Proponents of IP-over-WDM design point out (correctly) that failures in the IP layer cannot be completely overcome by recovery methods in the lower (optical/circuit) layer [5].

This subtle point is often overlooked and can be better explained with the help of Fig. 9. Consider two IP routers connected over the wide area by a switched optical network [Fig. 9(a)]. The latter can recover from any failures that happen *within* the optical network—for example, fiber, optical switch, or optical device related failures (all within the cloud). But if the failure is in the IP layer—for example, a router interface fails or the router fails entirely—then recovery cannot be accomplished in the optical network. Additional capacity (another IP interface or router) must be overprovisioned in the IP layer to deal with such failures. But dimensioned spare capacity (more interfaces, switching bandwidth) in the IP layer is agnostic to the *type* of failure. It can recover from both IP failures as well as optical failures. And so there is *no need* for recovery in the optical network—it is simply redundant here!

However, the problem with handling all recovery in the IP layer is that it comes at the high cost of much more expensive BRs and their core-facing ports. On the other hand, our choice of replacing BRs and using core-facing circuit ports has the important consequence that *all* recovery can be performed in the circuit layer [as shown in Fig. 9(b)].

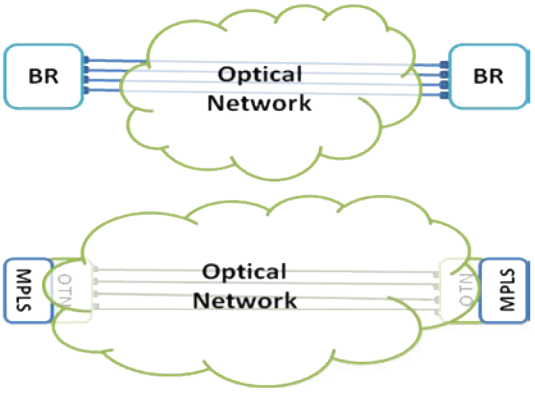

Fig. 9. Removing redundant recovery mechanisms.

There is *no difference* between the loss of a circuit interface or link as both result in the same failure backed up with spare capacity in the optical network.

2) Enabling a Fully Meshed IP Topology: We are not the first to suggest a fully meshed IP topology. In the 1990s, several ISPs created IP core networks that were fully meshed, with BR interconnections running over ATM virtual circuits. However, such construction suffers from a serious deficiency known as the $O(N^2)$ problem [15], which contributed to the failure of IP-over-ATM.

When N routers are connected by a full mesh, the distributed link-state routing protocol creates N routing adjacencies for every router. When a link goes down, the routers on both ends tell all their adjacent routers (the $N-1$ other routers in the network), and all of these routers tell their $N-1$ neighbors, resulting in $O(N^2)$ messages, each of which triggers recomputation of the shortest-path tree, and causes load, on the router CPU. If the extra load crashes the router the situation is worse: it generates $O(N^3)$ messages. Such a cascading sequence of events, in the worst case, can crash an entire network.

The $O(N^2)$ problem is an artifact of using distributed link-state routing protocols in a full-mesh topology. With SDN, our control architecture eliminates distributed routing protocols within a controller's domain. In our design, when a node or link fails, the affected switches inform the controller of the failure (at worst $\sim O(N)$) and failover to preprogrammed backup paths. The controller may choose to optimize routes by recomputing paths and downloading new flow entries, In either case the complexity remains $O(N)$. Note that precomputed backup paths are also possible in today's (non-SDN-based) networks; nevertheless the $O(N^2)$ issues remains (in a full mesh). Eliminating distributed routing in SDN-based networks makes possible a full-mesh IP topology.

3) Unified Control Over Packets and Circuits: IP topologies today are static for good reason. If we change the circuit topology, the routing protocol has to reconverge. Because the routing calculation is distributed across all routers, the outcome is hard to predetermine—a small change in the fiber links may require all the shortest-path trees to be recalculated and for packets to take very different paths. Operators are understandably reluctant to change the topology. It is not surprising that IP links today are always made static with no interaction between packet and circuit networks; as a result no unified platform exists today.

With DCS, packet routing changes much more often—the whole reason for dynamically changing circuits is to add/remove bandwidth on demand and reroute traffic around failures.

GMPLS was the only previous attempt to create a unified control plane (UCP) for packets and circuits, but because it was built on top of all the existing complex control planes, it proved too complex to use [16]. After a decade of standardization, there are no significant commercial deployments of GMPLS as a UCP [17,18].

With an SDN control plane, routing decisions are logically centralized and therefore converge much faster when bandwidth is added/removed or links fail. Logically centralized decisions (albeit by a distributed system of controllers) means backup paths can be precalculated. Circuits can be brought up and down without fearing network disruption. We can also *choose* which flows to effect and which flow routes to leave unchanged. In contrast to today's IP networks, none of these dynamic link changes need be disruptive to existing packet flows elsewhere in the network. The controller makes the decision of creating, modifying, or deleting a link, and it only affects the flows traversing that link.

IV. Prototype

We implemented our architectural approach in a proof-of-concept prototype [Fig. 10(a)]. It involved instantiating the flow abstraction across packet and circuit switches, for which we extended the OpenFlow protocol to build a *common switch-API* that manipulates the data abstraction of flow tables in both packet and circuit switches [19]. We implemented the API in hybrid switches—we use Ciena CoreDirectors, which have limited packet-switching capability, but are the precursors to the hybrid switches described in Subsection III.A and [20].

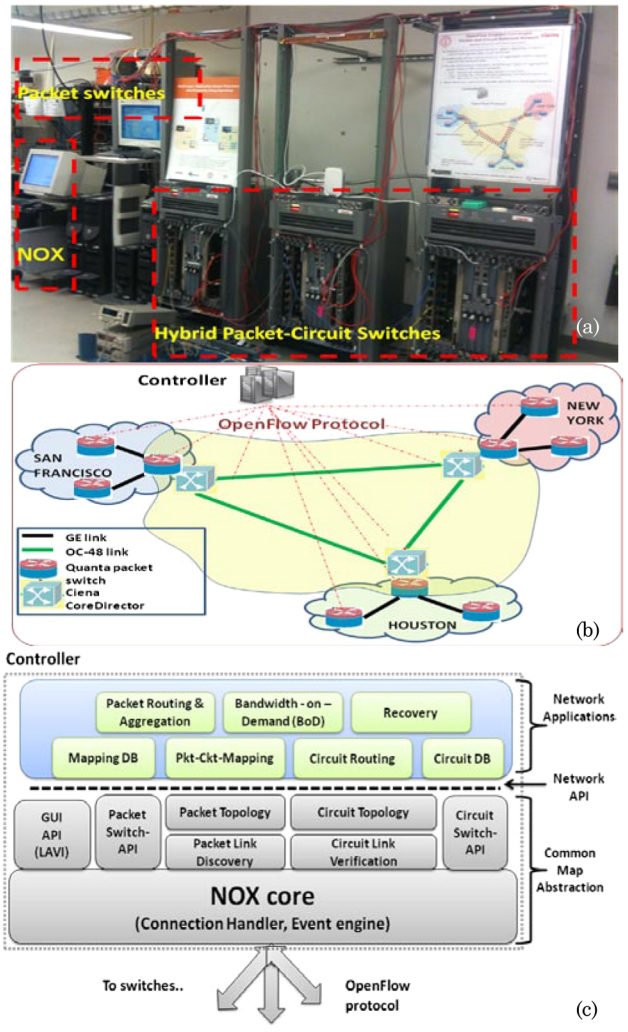

Fig. 10. (a) Prototype. (b) Emulated WAN. (c) Controller software architecture.

We also instantiated the common-map abstraction by implementing circuit-network related modules, such as discovery and (south-bound) circuit-switch APIs, in an existing NetOS (for packet networks) called NOX [21]. Together the packet and circuit modules in Fig. 10(c) present a (north-bound) network-API to applications for manipulating the common map. With the necessary interfaces implemented in the switches and the controller, we built a lab prototype that *emulates* wide-area network structure, similar to our proposal in Subsection III.A, i.e., ARs clustered in a city's PoP connected to hybrid switches [Fig. 10(b)]. Importantly, all switches in the network—standalone packet switches (ARs) as well as the hybrid switches—support the common switch-API. No other routing or signaling protocol is used, and all decision making is done by applications implemented in the controller, which have global view and full control of the network to jointly optimize network functions across packets and circuits.

A. Software Architecture

State in the hybrid switches is maintained by the lower-level applications. For example, the circuit-routing module is responsible for routing and maintaining the full mesh of circuits between PoPs in the fiber network. The Pkt-Ckt-mapping module is responsible for maintaining the mapping of ARs (and labels) to circuits in the hybrid packet-circuit switches. It inserts and updates matching rules in the packet part of the hybrid switches that identify incoming packets by tags representing aggregated packet flows to a destination AR. We used VLAN tags to represent the ARs instead of MPLS labels due to switch-ASIC limitations.

The higher-level applications include the packet-routing module, which is responsible for determining the BGP-next-hop for all IP-destination prefixes, and updating rules in the ARs. The bandwidth-on-demand (BoD) module monitors the circuit-flow state for their bandwidth usage in the hybrid switches and resizes the circuits according to application needs. Finally, the *recovery* module can preprogram backup paths for network failure scenarios for fast failover, and reactively optimize the backups on a more relaxed timescale after failover happens and the exact failure is known.

B. Evaluation and Discussion

We prototyped all of these applications on our prototype in fewer than 5000 lines of code, which is at least two orders of magnitude lesser code than an equivalent implementation using current industry-standard solutions [10,22]. Our code is not production ready, mainly because NOX is designed to run on a single server. More work would be

needed to ensure precise equivalence of function and reliability. But that would not come close to increasing the code by two orders of magnitude. A more detailed comparison is presented in Chapter 3 of [10]. We show in the next section that simplicity leads to lower costs.

More advanced NetOSes are commercially available and are designed to run on multiple servers for performance scale and resiliency [13]. The authors of [13] argue that compute/memory resources are not limiting factors for scale in a server cluster, but the consistency overhead for maintaining a network state can be. But such consistency need only be maintained on a per-network event timescale (tens of thousands/second), instead of a per-flow (millions/second) or per-packet (billions/second) timescale. And so maintaining (eventual) consistency for a (relatively) small number of events per second is what allows the controller to scale.

V. IP-AND-DCS MODEL

In this section, we perform a Capex analysis of a core network with the same PoP locations and TM as the one explored in Section II, but now with the design choices introduced in Section III. We outline the design methodology and present comparative Capex results.

A. Design Methodology

1) Topologies and Traffic Model: We use the same major PoP locations in the IP network that we used in the reference design [Fig. 1(a)]. The remote-AR locations are the same as well. The major differences are 1) BRs in the PoPs have been replaced by the hybrid switches, and 2) the edge topology from Fig. 1(a) is no longer used. Instead we use a fully meshed IP PoP topology. Since there are 16 PoPs, the IP topology has 120 edges. As before, each edge will be dimensioned into multiple, parallel links. The WDM topology remains the same as the one used in the reference design [Fig. 1(b)], and we use the same AR-to-AR unidirectional traffic model.

2) IP Edge Dimensioning: As before we first dimension for the traffic demand, then we account for recovery, and finally we overprovision for traffic uncertainties. In dimensioning for demand traffic, there is no need to run an SPF algorithm as every AR is one hop away from every other AR. We consider each PoP pair separately, and aggregate the demand from all the ARs in a PoP to ARs in the other PoP in both directions. We pick the largest aggregated demand and set it as the bidirectional demand for the PoP pair. Since this edge is actually realized by an OTN circuit, we calculate the number of time slots needed on the edge to satisfy the bidirectional demand assuming a minimum switching granularity of ODU0 (1.25 Gb/s) and the use of ODUflex to treat all time slots (even on multiple waves/interfaces) as part of the same "circuit."

To dimension for recovery, we use a simple shared-mesh-restoration algorithm, details of which are presented in [10]. Finally we dimension for traffic uncertainties by overprovisioning the total traffic and recovery demands by the same 4× overprovisioning factor we used in the reference design. Note that the AR to core-switch edges within the PoP are dimensioned exactly the same way as they were for the AR to BR edges in the reference design.

3) IP-PoP Dimensioning: After each edge is dimensioned in the full-mesh PoP topology, the number of parallel backbone links per edge is found by dividing the dimensioned time slots per edge (demand + recovery) by the number of time slots that fit in a 10G interface. Since we assumed that each time slot is an ODU0 (1.25 Gb/s) and eight of these fit in an ODU2 (10 Gbps), we can calculate the number of ODU2 interfaces (or rather OTU2 interfaces/waves) needed. Next the number of hybrid switches is determined by first figuring out the number of OTN switches (640 Gb/s switching capacity) required to satisfy the core interface—and the number of packet switches (also 640 Gb/s switching capacity) required to satisfy the access interfaces—and picking the greater of the two as the number of hybrid switches with 1.2 Tbps of switching capacity (with 640G packet and 640G circuit-switching capacity). Finally the number of 10G access links and ARs is determined by exactly the same procedure as the IP-over-WDM reference design.

4) WDM System Requirements: The final step of the design methodology is to route the circuits that make up the full-mesh core topology on the fiber topology. This procedure is exactly the same as the one followed for the reference design.

B. Capex Analysis

Circuit switches are much more scalable than packet switches, as they are simpler and more space efficient than packet switches of equivalent capacity. They are also available at a much lower price. Router ports cost 10 (or more) times as much as a circuit port with the same capacity. While some of the price difference is because router vendors enjoy higher margins, it also reflects higher part costs. Fundamentally, packet switches perform far more functions than circuit switches, and do so at a much smaller granularity, and at much faster timescales. Port costs, however, are difficult to obtain, and so, in our analysis, we use port price as a proxy for port costs.

Table II compares top-of-the-line commercial products—three types of circuit switches (based on fiber, wavelength, and time-slot switching) and an IP BR. The router consumes seven times the power (in W/Gb/s) and costs 10 times more (in $/Gb/s) than the TDM switch—and consumes 70 times the power and is 12 times the size (cubic-inch/Gb/s) of the WDM switch (details in [10]).

As before, all types of parts in our overall network cost, their exact usage, and their relative costs are described in detail in [10]. All costs are derived from the cost modeling in [6].

The core switches are hybrid switches with 640 Gb/s of switching capacity for both packet and circuit parts,

TABLE II
COMPARISON OF PACKET AND CIRCUIT SWITCHES [10][a]

Switch Type	Fiber Switch	WDM Switch	TDM Switch	Packet Switch
Switch example	Glimmerglass IOS600	Fujitsu flashwave 7500	Ciena CoreDirector	Cisco CRS-1
Switching capacity	1.92 Tbps	1.6 Tbps	640 Gbps	640 Gbps
Power	85 W	360 W	1440 W	9630 W
Volume	7″ × 17″ × 28″	23″ × 22″ × 22″	84″ × 26″ × 21″	84″ × 24″ × 36″
Price	<50	110.38	83.73	884.35

[a]Price values are in $1000s and are derived from [6].

resulting in a cumulative 1.28 Tbps of switching capacity. We deem the packet-switching part to be simpler than the ARs, with switching being limited to MPLS labels. As such they are similar to the switches being discussed as carrier Ethernet or MPLS-TP switches in [6], and so they are cost modeled the same way. The circuit-switching part is modeled with an OTN switch fabric and ODU2 (10G) interfaces from [6]. The AR and WDM system part costs remain the same as in Section II.

Figure 11 shows the overall results of our Capex analysis. With the design choices made in IP-and-DCS, port costs are reduced by using cheaper core-facing circuit ports, reducing their number by keeping all transit traffic in the circuit domain for normal and recovery scenarios, and removing redundancy in recovery mechanisms in the two layers by performing all recovery in the optical layer, while keeping overprovisioning levels the same and benefitting from on-demand bandwidth.

The overall number of core ports is reduced in IP-and-DCS (1480) when compared to the reference design (2564). As a result, we achieve nearly 60% in overall Capex savings when compared to the reference IP-over-WDM design. Most of these savings come in the backbone switches, which see an 85% reduction in cost (these include the backbone chassis and access and core-facing ports).

We also see a 25% reduction in WDM system costs (transponders and optical components). This reduction can be attributed to the design choices of full-mesh topology with shared-mesh restoration, which ultimately results in fewer 10G "waves" being required from the WDM network. In the reference design 1268 10G waves are routed in the WDM network. In our final design only 740 10G waves are needed, leading to fewer WDM systems and corresponding transponders.

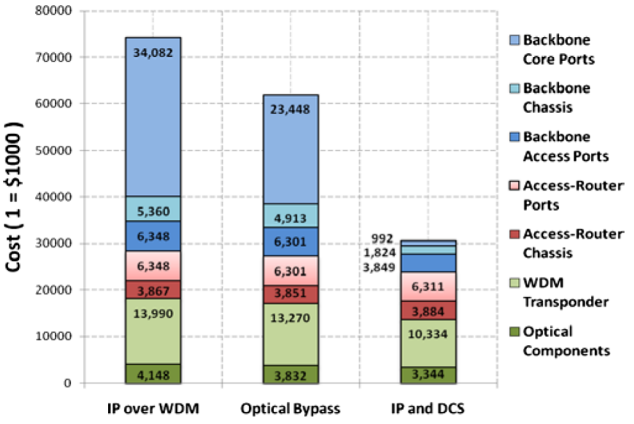

Fig. 11. Capex results.

It is also worth pointing out that our design achieves 50% in overall Capex savings when compared to the IP-over-WDM design enhanced by optical bypass (the middle column in Fig. 11 corresponds to the 50-edge case in Fig. 4). This is a direct result of keeping transit traffic in the circuit layer for normal and recovery scenarios using a dynamic-circuit layer instead of a static one.

1) Varying Traffic Matrices: The benefits of dynamicity in the optical network can be readily seen when we vary the TM. Figure 12 shows the results of our Capex analysis for three different TMs (same aggregate traffic demand of 2 Tbps). We show the traffic sourced by each PoP for three TMs in Fig. 12(a)—TM1 is the original TM (with peaks in NY and LA) that we have used in the analysis presented thus far. TM2 shows a more evened out distribution of traffic with smaller peaks, while TM3 is less balanced, like TM1, but with peaks in completely different cities (Chicago and Washington, DC).

Figure 12(b) shows the overall Capex results for each design with the three TMs. The Capex columns for TM2

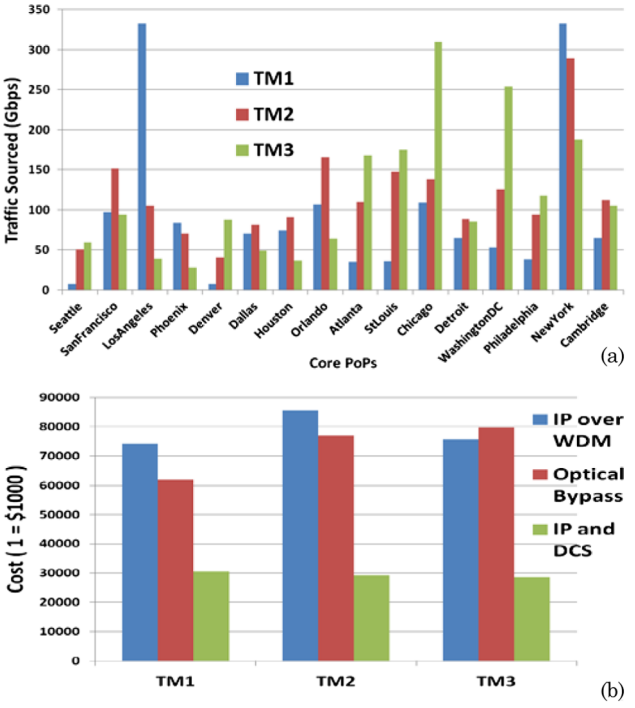

Fig. 12. (a) Three different TMs. (b) Effect of varying TMs.

and TM3 show different trends for the designs. For TM2, the TM is more evenly distributed. It results in more IP core ports in each PoP for both sourced traffic and transit traffic, which in turn results in higher Capex ($11 million more than for TM1). TM3, which is less balanced, results in reference design costs similar to TM1. But irrespective of the traffic distribution, the IP-and-DCS design yields nearly the same Capex costs for each matrix, reflective of how the design mirrors the matrix by reducing transit traffic and uses cheaper dynamic-circuit-switched core-facing ports.

Interestingly, for TM3 there is a complete erosion of savings from bypass—the static-bypass case is actually more expensive than the reference design. This is obviously a result of using the bypass candidates selected for TM1, in the Capex analysis for TM2 and TM3. It highlights a problem with static bypass.

With static bypass the design decision to create bypass is done offline and beforehand and then put into place. But if the TM changes significantly (as in going from TM1 to TMs 2 and 3), the static-bypass decisions cannot be changed—we are stuck with them. And so if bypass decisions are made-statically, and the TM can change, the IP network has to plan for such change, thereby reducing the savings from bypass. On the other hand, our IP and dynamic-circuit network design is insensitive to changes in the TM irrespective of how great the change may be.

2) Scaling Traffic Load: Finally, in Fig. 13 we show the effect of scaling TM1 to five times the original aggregate bandwidth demand. Scaling the TM is an effective way to plan for increasing traffic demand. When decisions are made to upgrade a backbone network with new purchases of equipment (Capex), they are often done with future traffic growth in mind. No one upgrades a network every year. Thus equipment is put in place such that the network can deal with increasing year-to-year traffic and upgrades that are disruptive to operation are not required for a few years.

So if we plan for 10 Tb/s traffic instead of the current 2 Tb/s, we see that our design affords nearly $200 million in savings. Importantly we find that the Capex costs are *diverging* with increasing traffic demand. Our design choices lead to a Capex versus demand slope of $11 million/Tb/s, which is significantly lower than the slope for IP-over-WDM with ($23 million/Tb/s) and without ($29 million/Tb/s) static optical bypass.

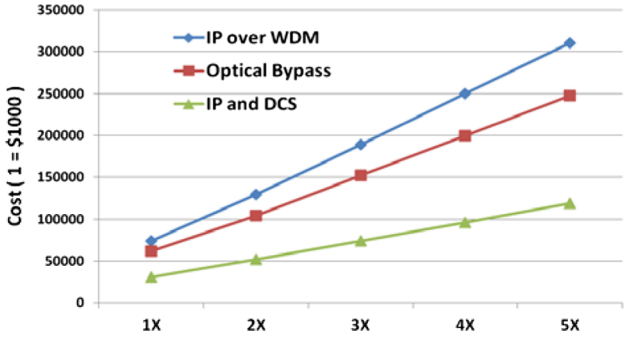

Fig. 13. Effect of scaling TM.

3) Opex Analysis We have also performed a limited operational expenditures (Opex) analysis accounting for power consumption, maintenance costs, and rack rentals. Our design achieves 37% cost savings compared to IP-over-WDM for TM1 (details in [10]).

VI. Related Work

Previous studies on core network Capex using packet and optical switching include [3–6]. One of the earliest works [4] makes similar conclusions on savings as our study. But our work differs in a number of ways. First, the authors of [4] make an erroneous assumption that the IP and fiber topologies are identical. Clearly this is not the case (as seen in Fig. 1 and also pointed out by [5]); the assumption results in significantly different dimensioned numbers for the optical network. Second, [4] neglects the redundancy scenario pointed out in Subsection III.C. It suggests the use of a dynamic optical network for recovery, but ignores that fact that the BRs in the IP network have to be dimensioned for recovery anyway (leading to redundancy), as the optical network cannot recover from BR failures. In contrast, we *replace* BRs with hybrid switches such that the dynamic optical network can recover from *all* failures, leading to significantly reduced cost. Finally, [4] was published nearly a decade ago when GMPLS was still popular. Indeed the authors suggest its use for interoperability between IP and the optical network. But as we pointed out in Subsection III.C, GMPLS has not succeeded for a number of reasons. In contrast, we have proposed a UCP based on SDN that solves many of the issues with GMPLS [16].

In our analysis we use more up-to-date parameters for the network topologies, TMs, and cost matrices. Our cost model, detailed in [10], is derived from the extensive cost modeling in [6]. However, the Capex analysis in [6] ignores the effects of recovery and overprovisioning, which are key elements of core network design. On the other hand, the study in [3] does consider recovery and overprovisioning, but only in the case of static optical bypass.

The only study that takes into account recovery, overprovisioning, bypass, dynamic-optical-switching, and all associated network costs is [5]. Our network modeling in Section II for IP-over-WDM networks follows a design methodology similar to [5]. However, [5] finds *no* cost savings in using packets and circuits together compared to IP-over-WDM. There are two main reasons. First, the authors acknowledge that recovery performed in both layers is redundant and leads to higher cost. This will always be true given scenarios in which the IP-core network is considered separate from the optical network. In contrast we proposed a converged network in which BR functionality is replaced by hybrid packet-optical switches, thereby resolving the redundancy issue and leading to lower costs.

The second (and possibly more important) reason why [5] shows no savings is that it takes into account a lot of other (non-IP) traffic that the optical network services (predominantly private-line traffic). As such, [5] performs a Capex analysis of not just the IP network but also the

entire underlying optical (transport) network. In contrast, we focus solely on the IP network and the part of the optical network that supports it, simply because we believe that in the future, the only traffic carried by the transport network will be IP traffic. Indeed another paper by the same authors seems to support this trend, where as much as 60% of AT&T's transport network directly or indirectly supports IP networks [23]. And so we care only about IP network scale and how it can benefit from DCS.

Ultimately, none of the previous works have proposed the combination of architectural elements discussed in Subsection III.A: namely, the replacement of BRs with hybrid packet-optical switches, the use of a dynamic full-mesh topology, and the adoption of an SDN-based control plane for joint control over packets and circuits. Additionally, we are the first, to the best of our knowledge, to highlight the problems that led to the failure of previous proposals and suggest architectural solutions for the same.

VII. Conclusions

While there have been many proposals for hybrid packet and circuit-switched backbone networks, we believe this is the first to 1) use an SDN control plane to control both packets and circuits, and therefore 2) allow the use of a low-cost, full-mesh optical network to serve as the core of an IP network. We also believe it is the first comprehensive cost analysis of a new approach.

Of course, there is much work to be done to persuade a conservative industry to reconsider the architecture of their networks. But the pressing need for network operators to reduce their capital and operational costs (or go out of business) is likely to force a serious reevaluation of how they build and operate their networks.

References

[1] Cisco CRS-1, 2011 [Online]. Available: http://www.cisco.com/en/US/prod/collateral/routers/ps5763/ps5862/product_data_sheet09186a008022d5f3.html.

[2] S. Elby, "Software defined networks: A carrier perspective," in *Open Networking Summit*, Stanford, CA, Oct. 2011.

[3] J. Simmons, *Optical Network Design and Planning*, Springer. 2008.

[4] S. Sengupta, V. Kumar, and D. Saha, "Switched optical backbone for cost-effective scalable core IP networks," *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 60–70, June 2003.

[5] G. Li, D. Wang, J. Yates, R. Doverspike, and C. Kalmanek, "IP over optical cross-connect architectures," *IEEE Commun. Mag.*, vol. 45, no. 2, pp. 34–39, Feb. 2007.

[6] R. Huelsermann, M. Gunkel, C. Muesburger, and D. Schupke, "Cost modeling and evaluation of capital expenditures in optical multilayer networks," *J. Opt. Netw.*, vol. 7, no. 9, pp. 814–833, Sept. 2008.

[7] S. Shenker, M. Casado, T. Koponen, and N. McKeown, "The future of networking and the past of protocols," in *Open Networking Summit*, Stanford, CA, Oct. 2011.

[8] N. Spring, R. Mahajan, and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *SIGCOMM*, Aug. 2002.

[9] A. Medina, N. Taft, K. Salamatian, S. Bhattacharya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," in *SIGCOMM*, Aug. 2002.

[10] S. Das, "Unified control architecture for packet and circuit network convergence," Ph.D. thesis, Stanford University, Stanford, CA, June 2012 [Online]. Available: http://www.openflow.org/wk/index.php/PACC_Thesis.

[11] "Interfaces for the Optical Transport Network (OTN)," ITU Recommendation G.709 [Online]. Available: http://www.itu.int/rec/T-REC-G.709/

[12] OpenFlow Specification [Online]. Available: https://www.opennetworking.org/.

[13] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievsky, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *OSDI*, Vancouver, BC, Canada, 2010.

[14] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, "Efficient distributed path selection for shared restoration connections," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 761–771, Oct. 2003.

[15] E. Osborne and A. Simha, *Traffic Engineering With MPLS*. Cisco, 2002.

[16] S. Das, G. Parulkar, and N. McKeown, "Why OpenFlow/SDN can succeed where GMPLS failed," in *European Conf. on Optical Communications (ECOC)*, Amsterdam, The Netherlands, 2012.

[17] M. J. Morrow, M. Tatipamula, and A. Farrel, "GMPLS: The promise of the next-generation optical control plane. Guest editorial," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 26–27, July 2005.

[18] C. Matsumoto, "Packet-optical stays out of control," 2011 [Online]. Available: http://www.lightreading.com/document.asp?doc_id=208072.

[19] OpenFlow Extensions for Circuit Switches (draft) [Online]. Available: http://www.openflow.org/wk/index.php/PAC.C#Experimental_Extensions_to_OpenFlow.

[20] Ciena 5400 series [Online]. Available: http://www.ciena.com/products/category/packet-optical-switching/.

[21] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: Towards an operating system for networks," *Comput. Commun. Rev.*, vol. 38, no. 3, pp. 105–110, July 2008.

[22] S. Das, Y. Yiakoumis, G. Parulkar, P. Singh, D. Getachew, P. D. Desai, and N. McKeown, "Application-aware aggregation and traffic engineering in a converged packet-circuit network," in *OFC/NFOEC*, Los Angeles, CA, 2011.

[23] A. Gerber and R. Doverspike, "Traffic types and growth in backbone networks," in *OFC/NFOEC*, Florham Park, NJ, 2011.

Saurav Das received a Ph.D. in electrical engineering from Stanford University in 2011.

Guru Parulkar is an Executive Director and Board Member at ON.LAB and ONRC Research and a Consulting Professor of Electrical Engineering at Stanford University.

Nick McKeown is a Professor of Electrical Engineering and Computer Science at Stanford University. He is also a Board Member and Faculty Director at ON.LAB and ONRC Research.