Material for lecture #4

Part 1. Scalability of resource reservations

This brief document approaches the scalability issue mainly from the viewpoint of bandwidth reservations. The starting point is a packet network with access and core routers and the issues to be evaluated are the implementability, manageability and usefulness of resource reservations. The big question is whether additional or more elaborate reservations can somehow improve network performance or user satisfaction, particularly in a large network.

Let us introduce as a study case a network with N_a = 450 access nodes each generating on average 100 Mbps traffic into the core network. Now without considering more about the nature of traffic, it might be useful to make resource reservations in the core, so far for an unknown reason. There could either be a reasonable number of core nodes, that is, from 5 to 20, or a large number of core nodes, say something like 100. In order to evaluate the implementation and manageability of reservations; let us take 4 scenarios:

A) N_C = 9 core nodes with pure routing

B) N_C = 9 core nodes with full mesh connectivity

C) N_C = 9 core nodes with full mesh connectivity between all access node pairs

D) N_C = 90 core nodes with full mesh connectivity

The networks are shown in figures 1 and 2.

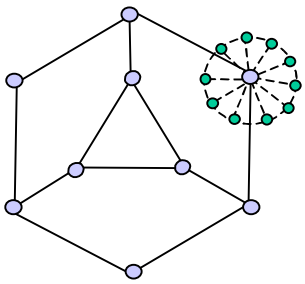

Fig. 1. A network with 9 core nodes

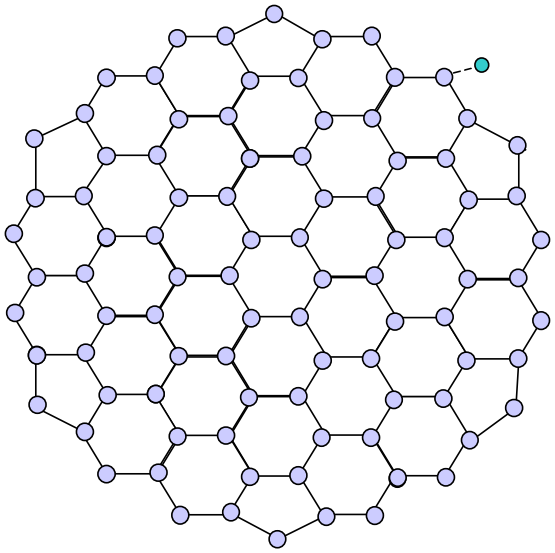

Fig. 2. A network with 90 core nodes

Let us define a connection as the entity that has a capacity reservation between two network nodes. In principle, an individual micro-flow could be a connection if the network is based purely on connections, but this approach is not discussed here. In contrast we assume that reservations are used between network nodes for aggregate traffic streams rather than for individual connections. A simple, but important question is, how many connections there are in the network, if reservations are used between each node pair on a certain network layer.

First, the number of core links (L) should be calculated, in our case $L(9)$ = 12 and $L(90)$ = 126. The average number of links attached to a core node is:

$$L_1 = 2L/N_C$$

Which produces the following figures: $L_1(9)$ = 2.67 and $L_1(90)$ = 2.80.

In addition we need to know the average number of hops between two core nodes, h. In our networks the average numbers are $h(9)$ = 1.83, and $h(90)$ = 6.95 (on the condition that traffic is divided evenly among all node pairs).

From these basic figures it is possible to calculate the total number of connections and the average number of connections on a core link. Interested readers may derive the necessary formulae. The results are anyway presented in Table 1. In addition the table shows the effort needed to manage the connections based on an (arbitrary) assumption that capacity and status of each connection has to checked once a day, and the required time for this operation is 1 min for each connection, and an additional 15 second for each link the connection traverses (these figures are not based on any real measurements).

Table 1.

Scenario	total number of connections (2-way)	average number of connections per core link	management of connections, hours/day
A (N_C=9, pure routing)	-	-	0
B (N_C=9, core mesh)	36	5.50	0.77
C (N_C=9, mesh between access nodes)	90000	13750	1912
D (N_C=90, core mesh)	4005	220.86	136

Someone may argue that 1 minute is not needed in practice, because everything can be automated. However, I am tend to think that if there is any manual operation related to the management of connections, the real figures are much longer. Indeed, I have personal experience on the management of an ATM trial network, and purely the configuration of all traffic parameters took several minutes, and if there was any doubt about the rights to make a reservation for a customer, the required time was manyfold. Thus, if any manual operation is needed, approximately fifteen nodes is the maximum size of the mesh (the overall situation should be controllable by one person, otherwise the management overhead increases rapidly). In contrast, scenarios C and D are possible only if manual operations are needed only in very rare cases.

As a result, even in a small network, there could be a significant overhead because of the management of the connections. Thus there must be good reason to introduce any reservation in a packet network on the whole. It seems that the general assumption is that there is a reason, though it is somewhat hard to define. Because the meaning of end-to-end reservation are discussed later in this lecture series, let us try to find the reason from the aggregate level. Maybe the answer is located somewhere in the area of resource management, overall congestion control or router performance. Perhaps.

This issue is, unfortunately, related to the dimensioning of links and connections, and therefore is an extremely complicated matter. What we need for our limited purposes, is an approximate relationship between average traffic and required capacity. The following relationship is not claimed to be generally applicable answer for the network dimensioning problem, rather it is rough formulation of the overall understanding of the author of this document:

$$C = A\left(1 + \left(\frac{A_0}{A}\right)^b\right) \qquad (1)$$

where A is the average traffic, C is required capacity, and A_0 and b are free parameters that can be used to tune the characteristics of the formula. Is it easy to see that for a average traffic $A = A_0$, the required capacity is 2 A_0. The factor b defines how quickly the effect of statistical multiplexing improves when the average traffic is increased. Theoretically, a selection of b = 0.5 is attractive (it is well justified if we assume that the traffic is an sum of independent random variables). However, that selection gives some peculiar results. For instance, if A_0 = 500 Mbps, 1 Mbps traffic requires capacity of 23 Mbps. In spite of the theoretical validity of this result, this kind of network dimensioning is not practical in reality. Therefore, we use in the

next evaluation a smaller value, b = 0.25. Note that a selection b = 0 means that there were no statistical multiplexing at all.

Table 2.

A, Mbps	max load
1	0.17
10	0.27
100	0.40
1000	0.54
10000	0.68

Now we can try to determine the effect of reservations by assuming that the capacity required by a connection is determined by the above dimensioning formula. In addition we assume that the reservation is constant and the required link capacity is the sum of the connection capacities (anything else is far for trivial and hard to implement without significant management overhead; in my mind, a statistical reservation, though widely used in reality, is not a real reservation[1]).

Table 3.

Scenario	average number of connections per core link	traffic per connection, Mbps	required capacity per connection, Mbps	required link capacity, Mbps
A (N_C=9, pure routing)	(1)	(3437)	(5560)	5560
B (N_C=9, core mesh)	5.50	625	1216	6688
C (N_C=9, access mesh)	13750	0.25	1.92	26452
D (N_C=90, core mesh)	220.86	5.62	22.9	5052

What can we learn from this example? First, scenario B might be feasible if it provides some other advantage not shown in Table 3. Second, scenario C is totally impractical as such, and consequently, a full mesh between access nodes definitely is not a scalable solution. Then if we compare scenarios B and D, the interesting result is that in a large network the link capacity can be only slightly decreased even though the number of nodes and links is tenfold - this is hardly a promising approach.

It appears that the number of nodes on the highest hierarchical layer in a network has to be limited into relative small number (maybe 15). This highest layer can make use of resource reservations, or permanent connections, whereas all traffic between two access nodes (located under different core nodes) usually has to traverse through 2 or more core routers without any own reserved resources.

This conclusion makes it possible to further consider the question of advantage of reservations or permanent connections. Namely, it might be possible to alleviate the

[1] You may think here for instance the reservation system used by airline companies

routing load by a more straightforward packet forwarding. Hence in a core network, an incoming packet is handled in the normal packet forwarding process in two routers, the first and the last core router, whereas in the intermediate routers a simpler forwarding process can be used. The question is, how big is this advantage?

The advantage depends solely on average the number of hops in core network, and can be expressed as follows:

$$gain = \frac{N_h - 1}{N_h + 1}$$

For a small network (like 9 nodes) this gain is relatively small (29%), and even for a large network (90 nodes) it is not very high (75%). But as discussed earlier, a large mesh is anyway unrealistic, so the advantage is unavoidably quite small. Note that since the mesh network with 9 nodes requires, according to our calculations, 20% higher link capacity than a pure routing network, it is not at all clear whether there is any attainable performance gain or cost saving (the final conclusion depends on how costs are divided between routing and link capacity).

Let us make the last notice by observing that if the network were built based on ATM technology (without routers), the effective consequence would be a scenario in which 9 ATM crossconnects form the core network, and 450 ATM access nodes were connected into these core nodes - but how?. Evidently, according to table 3, permanent connections between access nodes result in a undesirable situation: small bit pipes (1.9 Mbps) between access nodes, and low link utilization (13%). This is definitely a bad idea; without any routing capability, the only possible solutions seem to be either micro-flow reservations or to cope without any reservations. In IP terms, these approaches are end-to-end RSVP and best effort service, respectively. Fortunately, in IP we have the possibility to use packet prioritization and by that means to achieve something better.

Part 2. Scalability of WDM network

This part of the document is directly based on a conference paper "Wavelength Router as a Transport Platform for IP" by J. Kurki, K. Kilkki and A. Doria.

Wavelength routing has recently seen a remarkable upsurge in interest as a potential transport technology for IP traffic since the traffic demand of the Internet and other data services is growing exponentially. Due to the growth of traffic, primarily data traffic and the Internet, the networks started becoming exhausted in mid 1990's. Wavelength Division Multiplexing (WDM) offers a great solution for solving the fiber exhaustion. An important development was the Erbium Doped Fiber amplifiers that matured to make it possible to transmit initially 16 and today up to 200 wavelength signals simultaneously in one fiber over 600 km without requiring electrical regeneration. This has resulted in very big increase in the efficiency of transport and the simplification of the long distance network because a very large number of regenerators could be eliminated.

These systems are, however, point to point systems that need to be terminated at an electrical node that will typically be an IP-router in future networks. WDM technology can carry over 100 signals at 2.5 or 10 Gb/s. When these signals are terminated at an electrical node a very high switching capacity is needed. In many cases most of the traffic is transit traffic so it would be unnecessary to convert this to an electrical form. An elegant solution that provides sufficient bandwidth utilizes an optically transparent Wavelength Router. This is a device that consists of an all-optical switching core and interfaces. The idea of the wavelength router is similar to the IP router but the

switching is done in the optical domain. Networks could be built in a flexible way by adding new nodes while the control protocols dynamically configure the system.

An optically transparent wavelength routed metropolitan area network is depicted in Fig. 3. The client nodes are connected to their peer nodes by a signal at a wavelength between the Wavelength Routers (WR). The conversion of the client signal to a wavelength would be normally done at the WR-node but it could also be done at the client node. This would avoid the use of one transmitter / receiver pair but, since client nodes of many types are used, would impose strict control requirements to the wavelength to avoid shift to a wrong wavelength and resulting crosstalk.

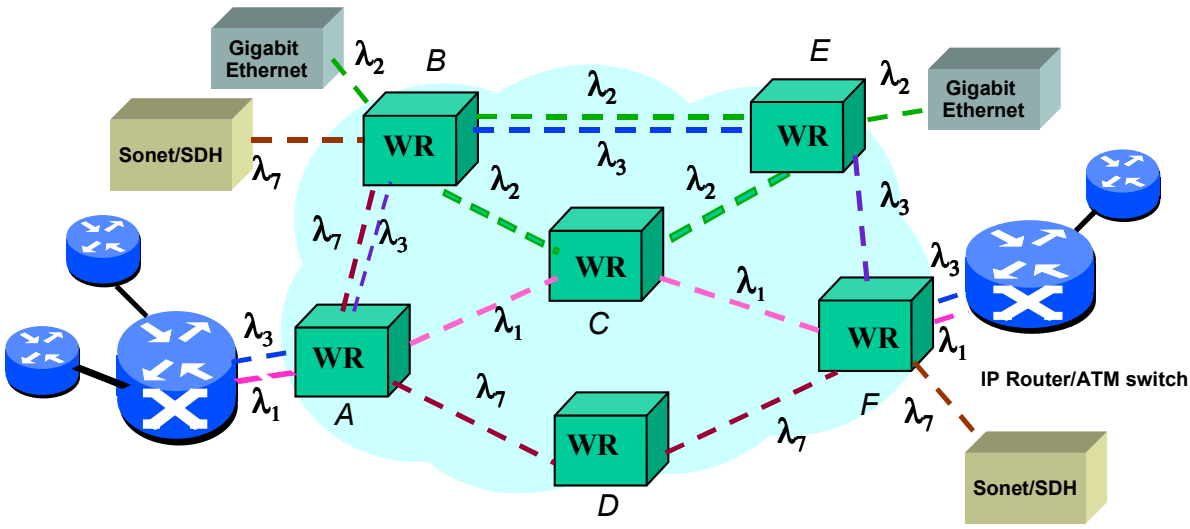

Fig. 3. Wavelength routing network consisting of Wavelength routers (WR) and a set of client nodes peers.

For the analysis, the network architecture in Fig. 3. is used. IP Core Routers are attached to the Wavelength Routers and IP Access Routers are then connected to the core router. The reason for this is that the granularity at the optical layer is very coarse (typically 2.5 or 10 Gb/s) and to achieve full connectivity to all the access routers a core router is needed for aggregation. Otherwise, the number of wavelengths connecting the Wavelength routers would need to match the number of all combinations down to access routers (as discussed in the beginning of this document).

A regular topology of a full mesh network was used to analyze the efficiency of the network. Assumption was made that there would be an IP router for traffic grooming connected to each of the Wavelength Router network nodes. We studied primarily the number of wavelengths needed to interconnect the nodes (*W*), the number of hops between IP Routers (*h*) that would be made in the optical domain before the traffic needs to be terminated in the IP routers. For a network topology in which each node is connected on average to 3 other nodes, the relationship between *h* and the number of nodes (*N*) is approximately as follows:

$$h \approx \sqrt{0.4N} \tag{2}$$

Based on this approximation the average amount of wavelengths needed on link between two optical nodes is

$$W = \tfrac{1}{3}(N-1)\sqrt{0.4N} \approx 0.2N^{1.5} \tag{3}$$

The results are shown in Table 3. Here relative IP router traffic = IP router traffic in a network with wavelength routing divided by the IP router traffic without wavelength routing = 2/(1+h). A wavelength routed network would comprise of 20 nodes connected with a direct link to three other nodes. Each link carries a minimum of 18 wavelengths. IP traffic would bypass two IP routers and thus savings in the IP layer capacity would be 50 %.

Table 4. Wavelength routing network dimensioning

Number of Optical nodes (N)	Average number of hops between IP- Routers (h)	Average number of wavelengths per link (W)	Relative IP Router traffic, %
5	1.4	1.9	83
10	2.0	6.0	67
20	2.8	17.9	52
50	4.5	73.0	37
100	6.3	208.7	27

Now we can use the same formula as in the first part of the document to dimension this network. This evaluation is base on the following choices: A_0 = 2 Gbps and b = 0.25. The main difference compared to the earlier evaluation is that the state of art in optical technology strongly favors a granularity of 2.5 Gbps. Therefore, in the following calculation, the final capacity for any traffic aggregate is rounded to the next larger multiple of 2.5 Gbps.

When formula (1) is used, the main task is to estimate the average traffic demand. If we make the assumption that the network is dimensioned based on the average busy hour traffic, then the evident parameter to be determined is the average traffic generated by a customer over that period. Although 5 kbps may seem to be small value for average traffic per customer, it means 2.25 Mbytes for every customer during one hour. Then if we assume that 15% of customers are active during the busy hour, the average traffic sent by an active customer is 15 Mbytes. This value actually appears quite large if the typical access rate is at most 64 kbps.

However, Internet traffic is growing exponentially, and what is now a reasonable estimation, could be a serious underestimation after a couple of years. Therefore, the following calculations are made also with an average traffic of 50 kbps and 500 kbps. These scenarios are possible if a majority of customers has a high-speed access with peak rate of 1 Mbps or higher. For illustration, 500 kbps corresponds to 1 million people with a continuos average access rate of 0.5 Mbps; or equivalently to 100 million users, the present amount of users in the Internet, at 5 kbps, all in *one* network.

Figure 4 illustrates the relationship between the number of optical nodes and the router interface capacity including interfaces to access routers (it is supposed that there are 1 million users and 100 access routers in total). With a full mesh, there is a point in which the capacity requirement of one router is minimized. With moderate traffic demand that minimum can be reached perhaps with 5 nodes, with high demand the minimum is around 10 nodes, and with extremely high demand it might be even 20 nodes. However, we must notice that this optimization concerns individual nodes, while the total capacity of all router interfaces in the network appears to increase practically always with the number of nodes when the total traffic demand is kept constant.

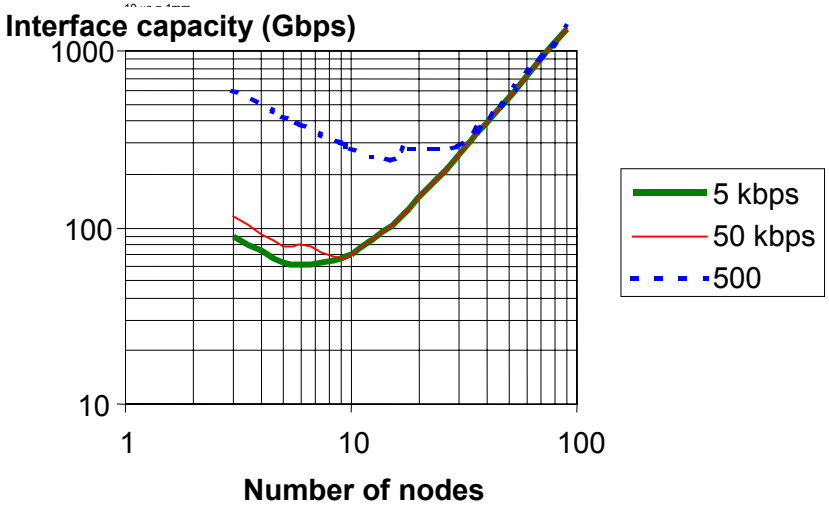

Fig 4. Total interface capacity of one router as a function of the number of core nodes. 5, 50 and 500 kbps = average traffic generated by one user, 1 000 000 users.

Part 3. Some thoughts about time scales

This part of the material provides some thoughts about relations between time scales and some very preliminary conclusions. The simple idea is to illustrate the huge difference between time scales that we need to assess in IP networks.

Everthing starts from IP packets: an IP router needs to be able to handle at least 100 000 packets per second. This means that packet handling time scale is in the order of 10 μs.

From buffering viewpoint the relevant time scale is of the order of 1 ms. Variations occuring on smaller time scales are not usually not critical because of the smoothing capacity of packet buffers.

From a human end-user viewpoint, the smallest relavant time scale is about 0.1 s, because shorter delays are not critical with any primary application. Then if the user takes any real action, e.g. because of bad service quality, it may easily take 10 seconds.

Then from network management viewpoint, any human interaction likely takes 15 minutes and in the worst case something useful happens only the day. Finally, if the situation is so bad that only capacity update is sufficient, the time scale could be 3 months.

The differences are illustrated in Fig. 5. However, the differences are so huge that it is really difficult to image any practical figure. Let us try to convert time to physical length. If the size of a packet is 1 mm, buffer size is 10 cm, and the smallest noticeable time scale is 10 m (=0.1 s). The end-user reactions are taken place only after 1 km, and network management will do something useful in the best case after 90 km, and more likely somewhere 9000 km away from the starting point (next day). If new a router or link is needed, that will happen when the process has reached a point somewhere far behind the moon.

One millimeter vs. the distance to the moon. If we need to have a way to the moon, should we handle every millimeter in a special manner. That does not sound very

reasonable. A more reasonable approach seems to be to concentrate on the bigger picture, that is, to the building of the highway.

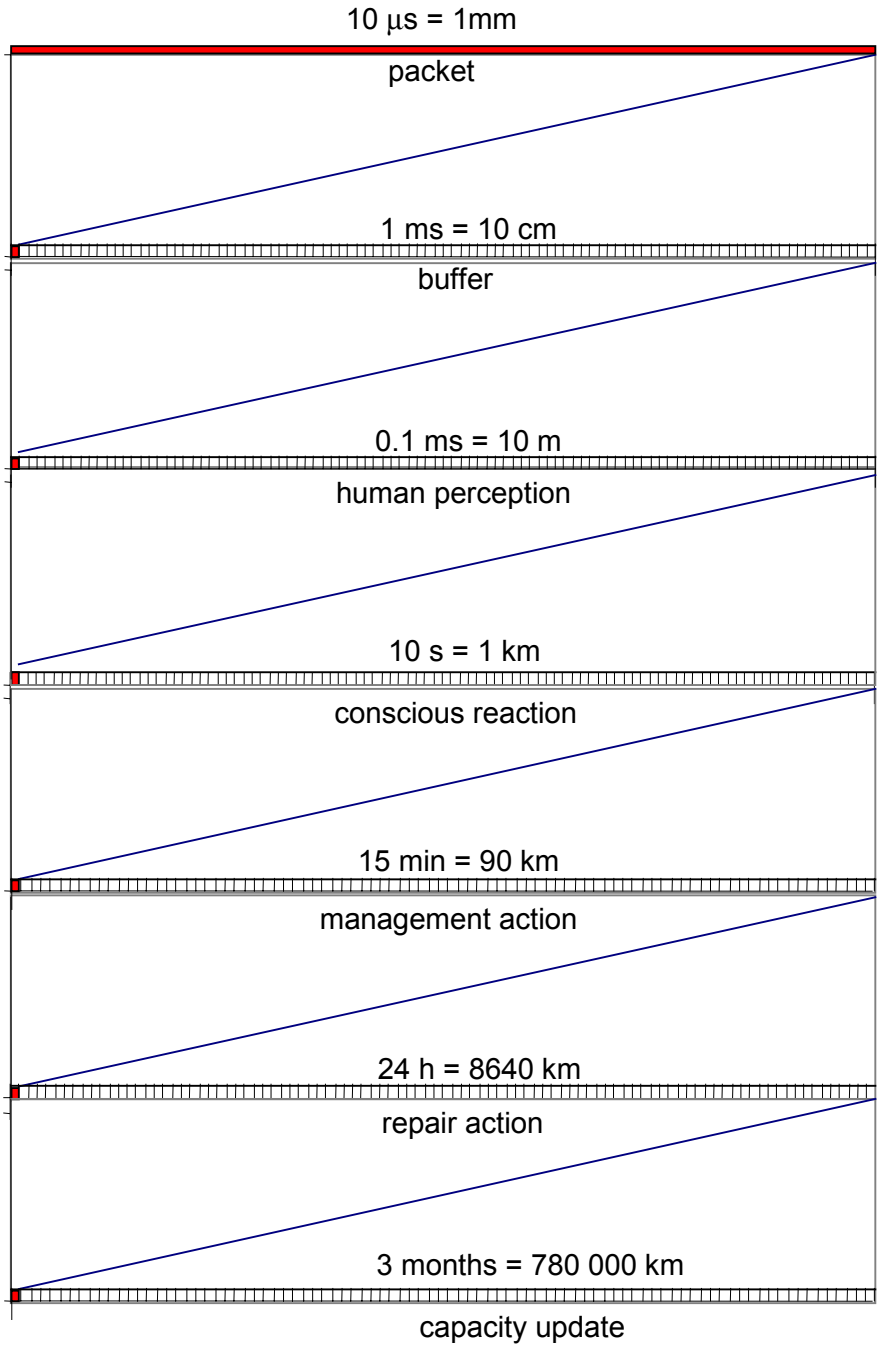

Figure 5. Timescales from packet handling to capacity update