# Quality of Service (QoS) Routing





# A Quick Review of NP-Completeness

- A problem is in Class **P** if it can be solved in polynomial time with a deterministic Turing machine.
  - Turing machine is a math modeling of stored-instruction computers.
- A problem is in Class **NP** if it can be solved in polynomial time with a non-deterministic Turing machine.
  - A non-deterministic Turing machine can simultaneously pursue an infinite number of computational paths.
  - It is an unrealistic but useful math model.

## P and NP Problems

- A Class P problem can be solved in polynomial time on real machines and is considered tractable.
  - Sorting, accounting, shortest path problems, spanning tree problems, and many other problems you use computers to solve daily
- A Class NP problem can be solved in exponential time on real machines.
  - You *may* be able to solve it in polynomial time.
  - All Class P problems are also NP.



- A problem in NP-P, if exists, cannot be solved in polynomial time on real machines and is considered intractable in practice.
- A good way to find a NP-P problem is to consider problems that do not have known polynomial solutions (algorithms).
  - map coloring problem, traveling salesman problem, automatic theorem proving, and some QoS routing problems
- However, no one can prove any of the above problems actually in NP-P.

## P=NP ?

- This one of the most fundamental, unsolved, problem in computer science.
- Most people believe it is too good to be true.
  - It would imply that all problems solvable by a non-deterministic Turing machine can be solved in polynomial time in real world.
- Still we have not mathematically proven it wrong.
- However, all most all computer scientists assume that NP-P is not empty.

CS 756 5

## NP-Complete Problems

- A problem X is **NP-Complete** if the following statement is true:
  - If we can solve X in polynomial time, then we can solve *all* NP problems in polynomial times.
- That is, if you come up with a polynomial-time algorithm for *any* NP-Complete problem, then you have proven P=NP.
- Since, we don't believe P=NP, we solve NP-Complete problems by devising
  - workarounds, or
  - approximation algorithms (heuristics).

CS 756 6

## The Routing Problem

- Traditional routing protocols (RIP, OSPF, etc.) mainly use hop counts to select paths.
- This does not meet the requirements of many emerging communication applications.
- For example, live multimedia applications must make sure that
  - Packet delays are bounded.
  - Jitters (changes in packet delays) are well controlled.



## Shortest Path Algorithms

- Given a graph G=(V,E), a shortest path algorithm finds a path with minimal distance, according to the given link costs, between a pair of source and destination.
- Shortest path algorithms are the foundation of network routing.
- Every real-world network routing protocol is either a centralized, distributed, or hybrid implementation of such algorithms.

- Dijkstra Algorithm:
  - $O(V^2)$, where $V$ is the number of nodes in the graph (or, routers in the network)
  - Used by OSPF
  - Works for non-negative link costs.
- Bellman-Ford Algorithm:
  - $O(EV)$, where $E$ is the number of links in the network
  - Suitable for distributed implementations
  - Used by RIP
  - Works for arbitrary link cost values (however, negative costs cannot form cycles)



# QoS Routing

- Find the path for a given source and destination that best satisfies a given set of criteria.
- Performance metrics include
  - Hop count
  - Delay
  - Jitter (the variance in consecutive packet separations at receivers)
  - Data loss rate
  - Available (residual) bandwidth
  - Queue length (available buffer space)

## Taxonomy

- For some metrics (e.g. bandwidth, buffer space), the state of a path is determined by the state of its bottleneck link.
- **Link-optimization routing** finds the path that "*optimizes*" the performance of its bottleneck link according to a given criteria.
  - Ex: bandwidth-optimization routing finds the path with the largest bandwidth in the bottleneck link



- **Link-constrained routing** finds a path whose bottleneck "*satisfies*" a given criteria.
  - Ex: bandwidth-constrained routing finds a path whose bottleneck supports the given bandwidth.

- For other QoS metrics, such as delay and jitters, the state of a path is determined by the combined state over all links of the path.
- **Path-optimization routing** finds the path that optimizes the given metric.
  - Example: delay-optimization routing finds a path with the minimum (accumulated) delay.
- **Path-constrained routing** finds a path that satisfies the requirement of the given metric.
  - Example: delay-constrained routing finds a path whose delay is bounded by the given value.



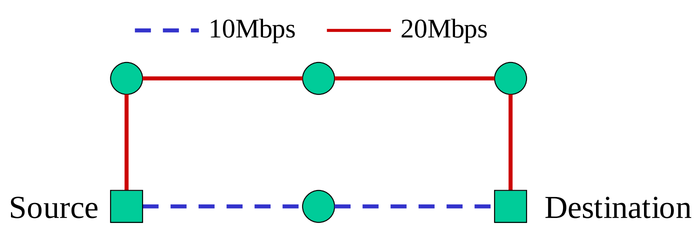
# Tractability

- Optimization problems can be solved by the traditional type of shortest path algorithms
  - Just use the given metric as the link cost.
- Constraint problems can be solved by their optimization counterparts.
  - To solve the delay-constrained routing problem, simply run the delay-optimization routing algorithm and see whether the best path meets the given delay bound.

# However …

- Using one metric is not enough for QoS applications.
- Focusing on one metric could results in waste of network resources.
  - Consider an application that needs 5 Mbps.
  - We run a link-optimization algorithm to find a path.






- Given the above network and link bandwidth, the algorithm will select the red path (whose bottleneck bandwidth is 20), rather than the blue path (whose bottleneck bandwidth is 10).
- However, the red path uses 5*4 Mbps of bandwidth, while the blue one uses only 5*2 Mbps.
- We need to consider a second metric, hop count.

## Composite Routing Problems

- ❑ Finding a path (between a source and destination) using more than one performance metric.
- ❑ Such problems can be derived from the above four basic routing problems.
- ❑ For example, the
  - bandwidth-constrained least-delay routing problem belongs to the
  - link-constrained path-optimization problem class.



## Polynomial Routing Problems

- ❑ Link-constrained, path-optimization routing
- ❑ Link-constrained, link-optimization routing
- ❑ Multi-link-constrained routing
- ❑ Link-constrained, path-constrained routing
- ❑ Path-constrained, link-optimization routing

- ❑ In the following, we will discuss important cases of the above problems and their solutions.

## Bandwidth-Delay Constrained Routing

- This is a case of link-constrained, path-constrained routing.
- It lends itself to multimedia applications that demand bandwidth availability and delay bound.



- Algorithm
  1. Eliminate all links that do not meet the bandwidth requirements.
  2. Run a traditional shortest path algorithm to find the minimum delay path.
  3. The path is accepted if it meets the delay constraint; otherwise report failure.

## Discussion

- We can always get rid of the "link-constrained" part by eliminating unsatisfactory links.
- The trick gives rise to the solutions for all the polynomial cases, except the last one, path-constrained, link-optimization routing.
  - We will not cover the case in this talk.
  - You are encouraged to figure it out on your own.



## NP-Complete Routing Problems

- Path-constrained, path-optimization (PCPO) routing
  - Example: delay-constrained, minimum-cost
- Multi-path-constrained routing (MPC)
  - Example: delay, delay jitter constrained routing

- **Notice**: There are two sufficient conditions for the NP-completeness of PCPO and MPC.
  - The two metrics are independent, and
  - they both use real numbers or unbounded integers as values



# A Precise Description of MPC

- A metric *d* is said to be additive if, given a path $P=L_1,L_2,\ldots L_n$, $d(P) = d(L_1)+d(L_2)+ \ldots +d(L_n)$.
  - The delay metric is additive.
- A metric *d* is said to be multiplicative if, given a path $P=L_1,L_2,\ldots L_n$, $d(P) = d(L_1)*d(L_2)* \ldots *d(L_n)$.
  - The transmission rate is multiplicative
- **Theorem**: Given any *N* additive/multiplicative metrics and their respective constraints, the problem of finding a path satisfying the *N* constraints is NP-complete.

## Shortest Path, Bounded Delay Routing

- The hop-count metric (where d(L)=1, for every link L in the network) is additive.
- The delay metric is also additive.
- Thus, the problem is NP-complete
- In general, finding the shortest path (in terms of hop counts) that satisfies an additive/multiplicative constraint is NP-complete.
- How depressing !



## Handling Delay and Jitter Constraints

- We've seen that delay and jitter constraints are difficult to deal with. They are, however, important to many multimedia applications.
- Fortunately, in networks that do provide QoS guarantees, additional help is available.
- Specifically, a network that guarantees QoS must use an advanced packet scheduling algorithm, such as WFQ.

- With many scheduling algorithms, delays and jitters of a traffic flow are functions of its allocated bandwidth.
- In general, the more bandwidth, the shorter the delay and the lesser the jitter.
- To solve a routing problem involving delay and jitter constraints, we first translate these constraints to a bandwidth requirement.
- We then face a bandwidth constrained problem, which is easy to deal with.



## Discussion

- The delay derived from bandwidth refers to the queueing and transmission delay at a router.
- It does not include signal propagation delays.
- Thus some distortions are introduced in using bandwidth to guarantee delays.
- For many communication links, this is ok.
- One important exception is satellite links, which cause extraordinary propagation delays.
- A practical solution is to simply exclude satellite links in delay-constrained routing.

## Present Incarnation

- Integrated Services, the service model that provides QoS guarantees to individual users and applications, do not have much momentum.
- Why do we bother with QoS routing at all ?
- QoS routing is emerging in a new form, termed **Constraint-Based Routing**, to route MPLS tunnels with performance criteria.



## QoS Routing in The Internet

- QOSPF: QoS Extension to OSPF
- OSPF LSAs are extended with bandwidth and link propagation delay.
- QOSPF uses a modified Bellman-Ford algorithm to perform shortest-path, bandwidth-constrained routing
  - this is an instance of link-constrained, path-optimization routing
- For a delay-sensitive flow, all satellite links are excluded in routing.